Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Estudios de Postgrado
Maestría en Artes en Tecnologías de la Información y la Comunicación

DISEÑO DE SOLUCIÓN TECNOLÓGICA EN LA NUBE PARA LA ENTREGA INMEDIATA Y CONFIABLE DE UN SISTEMA DE GESTIÓN DE OPERACIONES PARA FREELANCERS

Ing. Jorge Raúl Orozco Santolino

Asesorado por la Inga. Ma. Gabriela María Díaz Domínguez

Guatemala, septiembre de 2018

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

DISEÑO DE SOLUCIÓN TECNOLÓGICA EN LA NUBE PARA LA ENTREGA INMEDIATA Y CONFIABLE DE UN SISTEMA DE GESTIÓN DE OPERACIONES PARA FREELANCERS

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

JORGE RAÚL OROZCO SANTOLINO
ASESORADO POR LA ING. MA. GABRIELA MARÍA DÍAZ DOMÍNGUEZ

AL CONFERÍRSELE EL TÍTULO DE

MAESTRO EN ARTES EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

GUATEMALA, SEPTIEMBRE DE 2018

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Ángel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Oscar Humberto Galicia Núñez
VOCAL V	Br. Carlos Enrique Gómez Donis
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Mtro. Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Mtro. Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Mtro. Ing. Marlon Antonio Pérez Türk
EXAMINADORA	Mtra. Inga. María Elizabeth Aldana Díaz
SECRETARIA	Mtra. Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO DE SOLUCIÓN TECNOLÓGICA EN LA NUBE PARA LA ENTREGA INMEDIATA Y CONFIABLE DE UN SISTEMA DE GESTIÓN DE OPERACIONES PARA FREELANCERS

Tema que me fuera asignado por la Dirección de la Escuela de Estudios de Postgrado, con fecha 25 de agosto de 2017.

Jorge Raúl Orozco Santolino

ACTO QUE DEDICO A:

Dios	Por darme la vida y la motivación necesaria para culminar este ciclo.
Mi madre	Por todas sus enseñanzas, apoyo, paciencia y amor brindado en cada una de las etapas de mi vida.
Mi padre	Por todos sus consejos, por su esfuerzo y perseverancia para brindarme las mejores oportunidades.
Mis hermanos	Por ser ejemplos de vida, porque cada dificultad a la que se han afrontado, han sabido salir adelante.
Mi hermana	Por su cariño, atención y paciencia incondicional.
Mis amigos	Por brindarme su apoyo en las diferentes etapas de mi vida y todos esos momentos especiales compartidos.

AGRADECIMIENTOS A:

Inga. Maria Aldana Por su apoyo, dedicación y tiempo brindado durante el desarrollo del presente trabajo de graduación, ya que ha sido mi guía para completar cada una de las etapas requeridas.

Inga. Gabriela Díaz Por haberme brindado la oportunidad de asesoría para este trabajo de graduación, por su apoyo incondicional en cada uno de los entregables que me permitieron agilizar el proceso.

Ing. Estuardo Zapeta Por el apoyo brindado durante la realización del protocolo, por la motivación brindada para seguir adelante con el presente trabajo de graduación.

Mis compañeros Por el compañerismo, amistad y esfuerzo realizado en cada uno de los cursos de maestría, que permitió salir adelante y concluir la primera etapa de este camino.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

LISTA DE SÍMBOLOS

Símbolo	Significado
AWS	*Amazon Web Services*
PYMES	*Pequeñas y medianas empresas*
SSL	*Security Socket Layer*
SaaS	*Software as a Service*
CRM	*Customer Relationship Management*
NAT	*Network Address Translation*
RDS	*Relational Database Service*
AMI	*Amazon Machine Image*
HTTPS	*Hypertext Transfer Protocol Secure*
WAF	*Web Application Firewall*
ACL	*Access Control List*
VPC	*Virtual Private Cloud*
SSH	*Secure Shell*
RDP	*Remote Desktop Protocol*
AWS EFS	*Amazon Web Service Elastic File System*
DDoS	*Distributed Denial of Service*

GLOSARIO

Algoritmo	Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.
Aplicación	Conjunto de algoritmos preparado para una utilización específica ejecutado en un computador.
Arquitectura	Conjunto de diagramas que representan el diseño de un sistema en el cual se incluye la interacción de los diversos componentes.
Autenticación	Acción y efecto de autorizar el acceso a un sistema, mediante la validación de credenciales como usuario y contraseña.
Autoescalamiento	Acción y efecto de aumentar de forma automática el número de servidores que dan respuesta a las peticiones de los clientes.
Backend	Capa de una aplicación de *software* que identifica aquellos procesos que se ejecutan en el trasfondo de sus operaciones.
Balanceador de carga	Dispositivo que se encarga de la distribución de peticiones de los usuarios hacia diferentes servidores *web,* para mejorar el tiempo de respuesta.
Base de datos	Conjunto de datos que se encuentran relacionados entre sí y se almacenan de forma sistemática, para el uso de una aplicación de *software*.

Certificado de seguridad	Herramienta de validación y encriptación que se encarga de asegurar la información que se traslada entre un servidor *web* y el cliente.
Cloud Computing	Conjunto de tecnologías que permiten prestar servicios de computación como: servidores, bases de datos, almacenamiento, etc., a través de internet sin necesidad de alguna descarga.
Cross-site scripting	Inseguridad informática de una aplicación *web* que permite a un tercero inyectar código malicioso en un sitio *web*.
Disponibilidad	Cualidad o condición de encontrar disponible un sistema informático para su uso.
Encriptación	Corresponde al procedimiento de cifrado de información importante, mediante claves para impedir su legibilidad.
Escalabilidad	Cualidad que identifica la posibilidad de escalar de un sistema informático.
Firewall	Dispositivo físico o virtual que bloquea el acceso no autorizado a terceras personas.
Frontend	Capa de presentación de una aplicación de *software* que se encarga de la interacción directa con el usuario.
Hardware	Conjunto de dispositivos físicos que componen un equipo de computación.
Implementación	Proceso de configuración de una aplicación de *software* para su puesta en marcha.

Infraestructura	Conjunto de dispositivos e instalaciones que permiten la prestación de servicios informáticos.
Instancia	Corresponde a un servidor lógico alojado en *Amazon Web Services,* que se compone tanto de recursos computacionales virtuales como de sistema operativo.
Inyección SQL	Técnica de introducción de código malicioso, a través de los campos de formulario de una aplicación para explotar la base de datos.
Licencia	Convenio entre el dueño de una solución de *software* y el usuario que hace uso de ella cumpliendo ciertos requisitos legales y económicos.
Modelo *on-premise*	Fundamento utilizado para identificar aquel licenciamiento que requiere de la compra completa de un *software*, siendo este instalado y administrado por el cliente en sus servidores.
Máquina virtual	*Software* que simula un computador real con todos sus dispositivos para el procesamiento de programas.
Multi-tenant	Arquitectura de *software* que tiene como fundamento proveer de un servicio a múltiples clientes utilizando un solo servidor de *software*.
Procedimiento de aprovisionamiento	Proceso encargado de evaluar y suministrar más recursos a una solución de *software* de acuerdo a su demanda.

Protocolo de red	Conjunto de reglas que delimitan la comunicación en una red de computadoras.
Red	Conjunto de dispositivos relacionados entre sí, por un canal de comunicación que permite la transferencia de información, entre ellos.
Replicación	Proceso de copia y actualización de datos en un servidor de base de datos secundario, pudiéndose este alojar en instalaciones separadas al servidor primario.
Respaldos	Copias de seguridad de datos, realizadas a partir de datos originales, con el objeto de disponer de un medio de restauración en caso de pérdida de los datos originales.
Security group	Conjunto de reglas definidas en *Amazon Web Services,* para permitir el acceso a un servidor en puertos específicos.
Servidor web	Aplicación *web* encargada de recibir, procesar y responder a las peticiones de un cliente.
Software	Conjunto de componentes lógicos responsables de realizar tareas específicas ejecutadas en un computador.
Subdominio	Subgrupo de nombres de dominio clasificado como un dominio de segundo nivel, utilizado para acceder a un sitio *web* específico.

Subred	Subconjunto lógico de una red de dispositivos, creado con el propósito de facilitar su administración.
Tablas de ruteo	Archivo electrónico que almacena todas las rutas posibles entre los diferentes nodos de una red.
Variables de sesión	Conjunto de variables que identifican de manera única, la sesión de un usuario en una aplicación para agilizar su interacción.

RESUMEN

Los ideales de los trabajadores han cambiado durante los últimos años, pasando de buscar estabilidad y trabajos permanentes, trabajos que les brinden mayor flexibilidad de horario, independencia y oportunidad de aprendizaje. A este nuevo modelo de trabajo, se le conoce como *freelancing*, en donde los trabajadores laboran de forma independiente para diversas empresas y/o industrias. Por tal razón, se hace conveniente proveer de una solución tecnológica lo suficientemente flexible para adaptarse a cualquier modelo de negocio en cualquier industria.

En este estudio, se presenta el diseño de una solución en la nube como *Software as a Service*, que sea capaz de proveer un sistema para *freelancers.* Este diseño permite confirmar, mediante el uso de tecnologías en la nube como *Amazon Web Services*, y utilizando controles de seguridad en cada una de las capas de la solución, se puede prestar un servicio tecnológico que, mediante diversos modelos de suscripción sea accesible para los diversos *freelancers*.

Para esto, en el estudio se plantean tres objetivos fundamentales, siendo el primero de ellos un enfoque en los procedimientos de aprovisionamiento de recursos para los nuevos clientes. Como parte de dicho objetivo, se busca identificar todos aquellos procedimientos o métodos, los cuales se pueda responder de forma inmediata a la asignación de recursos que permitan satisfacer la demanda de los clientes, prestando de esta manera un servicio de *software* ininterrumpido y estable.

Los procedimientos de aprovisionamiento identificados se enfocan en la evaluación del rendimiento promedio de los servidores incluidos en la infraestructura de la solución, se procede con las acciones correspondientes, que

siendo el caso de un rendimiento bajo, se añadirían más servidores a la solución para mejorar la experiencia de usuario y evitar cualquier interrupción del sistema. A este proceso se le denomina como autoescalamiento, que utilizando los servicios de *Amazon Web Services,* se puede automatizar para disminuir el riesgo de una caída en la solución.

Adicionalmente, la solución considera que el servicio se prestará a los clientes, a través de la creación de un subdominio que estará directamente relacionado con él, de tal manera que, el cliente podrá reservar su subdominio y utilizarlo para futuros accesos a la solución. Por lo tanto, se identifican procesos que permitan evaluar la disponibilidad de un subdominio para su correspondiente asignación al cliente que está creando su cuenta.

Por otro lado, el segundo objetivo del estudio se enfoca en la identificación de todos aquellos controles de seguridad que permitan entregar una solución más confiable a los clientes. Para ello, se han considerado cada una de las capas de una solución de *software*: comunicación, aplicación, servidores y bases de datos.

En cada una de ellas, se identificaron los elementos, procesos y mejores prácticas que permiten que la solución de *software* sea más segura en cuanto a la información almacenada y a la disponibilidad del servicio. De esta manera, se asegura que la información registrada por los usuarios de la aplicación, resida en un ambiente con bajas probabilidades de sufrir algún ataque informático, ya que se reducen las vulnerabilidades de la solución, y al mismo tiempo se asegura que el usuario sea capaz de acceder a su información en todo momento, dada su alta disponibilidad.

Finalmente, se realiza un análisis de las diferentes variables que pueden llegar a impactar en los costos de la solución, con el objetivo de identificar el modelo de negocio más rentable que permita a *freelancers* hacer uso de la misma. Llegando de esta manera a determinar que el modelo de negocio más

rentable y efectivo para los clientes es, el modelo de *Software as a Service*, mediante una suscripción mensual pagada por los clientes, se permite el uso de la solución. Este modelo tiene muchas ventajas respecto a una solución *on-premise,* ya que los costos de administración y mantenimiento de la solución, se reducen en gran manera representando beneficios económicos para el cliente.

Además, se definen los planes de suscripción para ajustarse a cada uno de los tipos de *freelancers* que pueden presentarse, desde una persona individual, hasta una pequeña oficina. En cualquier de los casos, se plantea un costo de suscripción atractivo y accesible para que la probabilidad de captar clientes sea más alta. Teniendo el modelo de negocio y planes de suscripción identificados, se hace el análisis correspondiente para determinar si el negocio es rentable, a través de indicadores financieros como el margen bruto y la tasa interna de retorno.

PLANTEAMIENTO DEL PROBLEMA Y PREGUNTAS DE INVESTIGACIÓN

Problema social

Desde inicios de la presente década, la fuerza laboral está cambiando, presentando una fuerte tendencia hacia la independencia, en la que las personas optan más por un trabajo independiente que por un trabajo a tiempo completo. En el año 2014, se contaban 53 millones de *freelancers* (trabajadores independientes) en Estados Unidos, que representaban el 34 % de toda la fuerza laboral del país; en la Unión Europea al año 2013, se tenían 8.9 millones de *freelancers*, presentando un crecimiento de 45 % desde el 2004 al 2013. La fuerza laboral está cambiando rápidamente y no existen sistemas de información que les permita a los *freelancers* gestionar sus operaciones de forma fácil, segura y adaptable a cualquier tipo de industria en la que estén involucrados.

Soluciones tecnológicas y debilidades detectadas

Actualmente existen algunas soluciones tecnológicas que se enfocan en la gestión de operaciones de pequeños negocios, PYMES, permitiéndoles la identificación de actividades a sus proyectos/casos, identificación de los involucrados y de gastos incurridos en cada una de ellas; facilitando la administración de su negocio; sin embargo, no son completamente adaptables al mercado de los *freelancers*, o bien, algunas de estas soluciones están orientadas a industrias específicas, por ejemplo:

- MyCase, Clio, Abacus, Cosmo Lex: soluciones enfocadas en la industria legal.
- Salus, QuicDoc, PsytechSolutions: soluciones enfocadas en la industria de medicina.

Por otro lado, la mayoría de estas soluciones tienen costos demasiado altos, que resultan ser inaccesibles para los *freelancers*, o incluso, muchas veces para los pequeños negocios. Debido a que el mercado objetivo de las soluciones mencionadas previamente son las PYMES, muchas veces ellas tienen como prioridad la seguridad de la información, resultando en implementaciones en sitio sobre los servidores de los clientes, aumentando los costos de los mismos, ya que eso implica la compra de hardware, compra de *software*, mantenimiento y administración de los servidores, conlleva así contratación temporal de personal calificado.

Problema tecnológico

Dadas las circunstancias anteriores, la implementación de una solución para un *freelancer* o un pequeño negocio, resulta ser más compleja de lo que debería y de lo que ellos quisieran que fuera, obligándolos a utilizar herramientas no convencionales para gestionar sus operaciones, por ejemplo, Excel, Access u otras herramientas. Por lo cual, en algún momento los puede llevar a la descentralización de la información, a la pérdida de información valiosa, y a la falta de seguimiento de sus clientes.

Siendo todo esto ocasionado por la falta de herramientas tecnológicas que ofrezcan soluciones adaptables para las diferentes industrias de los *freelancers*, confiables (mediante el uso de técnicas informáticas como cifrados SSL y subdominios), y fáciles de instalar/configurar para su uso inmediato.

Pregunta central

¿Cómo se puede crear una solución en la nube que permita la entrega inmediata y confiable de un sistema de gestión de operaciones para *freelancers*?

Preguntas auxiliares

- ¿Cómo se puede aprovisionar computación, memoria, *networking* y almacenamiento en disco que permita la implementación inmediata y 100 % funcional de un *software* como servicio?
- ¿Qué componentes son necesarios para la creación de una infraestructura en la nube que cumpla con estándares de seguridad?
- ¿Cuál será el modelo de negocio a utilizar que capte la mayor cantidad de usuarios para crear una solución rentable?

OBJETIVOS

General

Diseñar una solución tecnológica en la nube que permita la entrega inmediata y confiable de un sistema de gestión de operaciones para *freelancers*.

Específicos

1. Diseñar los procedimientos de aprovisionamiento de computación, memoria, *networking* y almacenamiento en disco que permita la implementación inmediata y 100% funcional de un *software* como servicio.

2. Identificar los componentes necesarios para la creación de una infraestructura en la nube que cumpla con estándares de seguridad.

3. Definir el modelo de negocio a utilizar que capte la mayor cantidad de usuarios, para crear una solución rentable.

MARCO METODOLÓGICO

Tipo de estudio: cualitativo

El estudio tendrá como variable principal el diseño de la solución, que requerirá de investigación tecnológica tomando como base soluciones existentes en la industria, las cuales demuestren que el uso de tecnologías en la nube permite el aprovisionamiento de recursos de forma automatizada y cumplimiento de estándares de seguridad. Así mismo, se realizará un análisis FODA de diferentes modelos de negocio, que permitirán diseñar un modelo rentable.

Diseño: no experimental

La investigación tecnológica se realizará mediante el estudio y análisis de documentación, publicaciones y *whitepapers,* realizados por empresas que brinden servicios de tecnología en la nube, destacando las mejores prácticas utilizadas por las mismas.

Alcance: descriptivo

El resultado de la investigación expondrá los recursos, procedimientos y componentes que deben ser considerados para la entrega de una solución tecnológica en la nube de forma automatizada, logrando una implementación inmediata y segura. De igual manera, expondrá el modelo de negocio a utilizar, que será determinante para la captación de clientes, según el mercado objetivo.

Variables

Para llevar a cabo el análisis del presente estudio, se han definido una serie de variables, como se observa en la Tabla I, que permitirán medir la efectividad

de los procesos de aprovisionamiento, la eficiencia de los estándares de seguridad y la rentabilidad del negocio.

Tabla I. **Variables de estudio**

Variables	Definición	Sub-variables	Indicadores
Diseño SaaS	Diseño de solución tecnológica en la nube que permitirá brindar un sistema de gestión de operaciones a *freelancers*.	Aprovisionamiento de recurso	• Porcentaje de efectividad de los procedimientos de aprovisionamiento.
		Componentes de seguridad	• Porcentaje de cumplimiento de estándares.
Modelo de negocio	Estrategia de negocio que se utilizará para comercializar la solución.	Rentabilidad	• Tasa de rentabilidad • Margen bruto
		Ventaja competitiva	• Costo de oportunidad de los recursos empleados para la provisión del servicio.

Fuente: elaboración propia.

Técnicas de recolección de información

- Estudio de publicaciones y *whitepapers* sobre SaaS.
- Estudio de documentación técnica, tanto de *software* como hardware, elaborado por empresas que posean soluciones tecnológicas en la nube.

- Análisis de modelos de negocio de SaaS enfocados en pequeñas y medianas empresas.

Fases del estudio

Tabla II. **Fases del estudio**

Fase	Actividad	Descripción	Duración (semanas)
Aprovisionamiento de recursos	Investigación	Estudio de diversas fuentes de información que soporten el aprovisionamiento de recursos en la nube.	1
	Identificación de mejores prácticas	Análisis e identificación de mejores prácticas para los procedimientos de aprovisionamiento de recursos.	1
	Diseño de procedimientos de aprovisionamiento	Definición de los procedimientos automatizados de aprovisionamiento de recursos en la nube.	1

	Elaboración de documentación	Redacción de los procedimientos de aprovisionamiento.	1
Componentes para cumplimiento de estándares de seguridad	Investigación	Estudio de diversas fuentes de información que describan estándares de seguridad en solución tecnológicas.	1
	Identificación de componentes clave para estándares de seguridad	Análisis e identificación de componentes clave para el cumplimiento de estándares de seguridad.	1
	Diseño de infraestructura	Diseño de infraestructura de solución para el cumplimiento de estándares de seguridad.	1
	Elaboración de documentación	Diagramación de diseño y definición de componentes clave de infraestructura.	1

Modelo de negocio	Investigación	Estudio de diversos modelos de negocio de SaaS.	1
	Identificación de modelos de negocio exitosos	Análisis e identificación de variables que favorecen a un modelo de negocio de SaaS.	1
	Diseño de modelo de negocio rentable	Creación de modelo de negocio para una solución SaaS con sus variables.	1
	Elaboración de documentación	Descripción de las variables del modelo de negocio.	1

Fuente: elaboración propia.

INTRODUCCIÓN

Actualmente el uso de soluciones *web* ha incrementado drásticamente, debido a diversos factores que han facilitado la entrega de las mismas, como el acceso y velocidad del internet, y la mejora en tecnologías *web*. Esto ha generado un giro radical en el uso de aplicaciones empresariales, ya que muchas de ellas han empezado a migrar de soluciones cliente-servidor a soluciones *web*, y no solo soluciones *web* que se entregan, mediante una intranet para su uso privado, sino de soluciones *web* que se entregan, a través de internet para el uso público. Estas soluciones son conocidas como *Software as a Service* (SaaS), ya que corresponden a soluciones de *software* que se pueden utilizar como un servicio, este conlleva una suscripción, normalmente mensual, para hacer uso de la aplicación.

En el presente documento, se lleva a cabo un estudio detallado de los aspectos tecnológicos clave que se deben considerar para la implementación de una solución SaaS enfocada en la gestión de operaciones de *freelancers*, y permitir la entrega de una solución completamente funcional y segura de forma inmediata, siendo esta una de las premisas para toda solución SaaS.

El estudio estará estructurado en los siguientes capítulos:

El capítulo uno del estudio corresponde a los antecedentes. En este capítulo, se evaluará la evolución de los *freelancers,* desde la concepción del término hasta el incremento de los mismos en la actualidad. Además, se presentará la evolución en los modelos de entrega de sistemas de información empresariales, que inicialmente se realizaba mediante un modelo *on-premise*, el cual presentaba algunas desventajas respecto al modelo *Software as a Service*.

El capítulo dos corresponde a la justificación del estudio, que indica la razón por la cual se realiza. Esto considera la problemática actual que viven los *freelancers*, lo que conlleva a la búsqueda de soluciones tecnológicas que posibiliten la resolución de dicha problemática, siendo este documento el resultado del análisis y estudio de un diseño para la entrega de una solución SaaS, que cubra las necesidades de los *freelancers*.

El capítulo tres detalla los respectivos alcances del estudio desde tres perspectivas diferentes: investigativa, técnica y de resultados. Estos se encargan de definir qué resultados se presentarán como parte de la investigación ejecutada para identificar los procedimientos y estándares de seguridad que se deben entregar en la solución SaaS. De igual manera, detallará los resultados de efectividad de dichos procedimientos y estándares de seguridad, y a su vez, presentará un análisis de la rentabilidad y costo de oportunidad de la solución planteada.

El capítulo cuatro corresponde al marco teórico del estudio, en el que se detallan aspectos fundamentales de una solución SaaS, como lo es su arquitectura *multi-tenant,* la seguridad a aplicar a nivel de almacenamiento de datos, de aplicación y de infraestructura; y los modelos de negocio existentes. Además, se realiza una comparación de los aspectos anteriormente mencionados, indicando las ventajas y desventajas de los mismos, para proceder en el capítulo posterior a su respectiva evaluación.

El capítulo cinco que corresponde a la presentación de resultados, contendrá el detalle, definición e identificación de cada uno de los elementos para la solución SaaS. En esta sección se consideran tres aspectos importantes para el diseño de una solución SaaS: el aprovisionamiento de recursos de computación, el cumplimiento de estándares de seguridad y la definición del modelo de negocio. En este capítulo, se detallan dichos aspectos tecnológicos que permitan entregar una solución escalable y confiable, además de una

solución que esté al alcance económico de *freelancers* para asegurar que se tendrá una cartera de clientes amplia que den lugar a un crecimiento constante.

En el capítulo seis, se realizará el análisis de los resultados obtenidos durante la evaluación de los aspectos tecnológicos: efectividad de los procedimientos de aprovisionamiento, cumplimiento de los estándares de seguridad; y de los aspectos de negocio: tasa de rentabilidad y costo de oportunidad de los recursos. Esto ayudará a determinar, si el diseño planteado en el presente estudio es óptimo para una solución SaaS, o bien, si requiere de alguna mejora.

1. ANTECEDENTES

1.1. Evolución de *freelancing*

El término *freelance*, apareció inicialmente en una novela de Sir Walter Scott en 1819: Ivanhoe. En la cual, se hacía referencia a *freelance*, a aquel mercenario de la época medieval, que prestaba sus servicios de guerrero a cualquier persona que lo contrataba, sin importar la ética y moral. Básicamente utilizaba sus habilidades para desempeñar una labor para la cual había sido contratado. (Doriwala, J., 2014)

En los años de 1970, con la ayuda de la tecnología, los trabajadores empezaron a palpar grandes oportunidades para realizar sus tareas de forma remota, haciendo entrega de su trabajo, a través de medios electrónicos, siendo posible por el uso de las computadoras y las telecomunicaciones. (Anthony, Blau, 2002)

En años siguientes, las empresas empezaron a hacer el uso del teléfono para la entrega de trabajo en el área de tecnología, que permitió un ahorro del 30 % de los costos y un aumento en la productividad de 40 %. Así las empresas pueden observar que los trabajadores se veían beneficiados por la flexibilidad de horarios, además de la reducción de horas no productivas, como la movilización desde sus hogares hasta la oficina, lo cual representaba ahorro en tiempo y dinero para los trabajadores. (Blau, 2002)

Con el avance de la tecnología, diferentes industrias optaron por la contratación de *freelancers,* ya que les representaba un reducción en los costos. Algunas de las industrias que empezaron a utilizar el modelo de contratación de *freelancers* son:

- Periodismo
- Telemarketing
- Tecnologías de la información
- Diseño *web*

1.2. Inicios de *on-premise* y transición a SaaS

En los inicios de la era tecnológica, las soluciones de *software* se ofrecían en un modelo *on-premise*, que demandaba la instalación del *software* en el hardware de los clientes, mediante la compra de una licencia que les concedía el derecho de uso de la solución. Adicionalmente, el mantenimiento y administración de las soluciones de *software*, eran responsabilidad de los clientes, quienes debían velar por el correcto funcionamiento del hardware y brindar una alta disponibilidad del *software* a sus usuarios.

Cuando se dio inicio a la era del internet, la entrega de soluciones de *software* empezó a presentar un cambio, pasando de brindar soluciones *on-premise* a soluciones SaaS. Este cambio no fue tan notorio en los primeros años, debido a las diferentes limitantes que suponían el uso de soluciones *web*, como por ejemplo: la velocidad de internet, las tecnologías *web*, la comunicación *web*, entre otras. A finales de los años 90, muchos aspectos relacionados al internet y al uso de tecnologías *web*, presentaron mejoras de gran valor que abrieron puerta al uso de soluciones SaaS. Ganaron mayor impacto cuando los usuarios de internet aumentaron. En ese momento, las soluciones ofrecidas por internet crecieron y los usuarios empezaron a hacer uso de las mismas.

En ese tiempo, las soluciones SaaS fueron más utilizadas en pequeños y medianos mercados, en donde eran consideradas como una alternativa para problemas de corto alcance, las cuales les permitían cubrir ciertas demandas sin necesidad de realizar una fuerte inversión. Con este interés en juego, las grandes empresas de *software* vieron una oportunidad latente en el modelo de negocio,

en el que los usuarios solo pagaban por el uso de las funcionalidades que realmente necesitaban y por el uso que hacían de la solución. (Hossain, S., 2012)

El interés demostrado por los usuarios y las mejoras en el internet, originaron un incremento en las soluciones SaaS ofrecidas y en el uso de las mismas. Grandes empresas de *software* iniciaron su incursión en el modelo de negocio, ofreciendo cada vez soluciones más robustas y de mayor alcance. A su vez, los usuarios comprendieron los beneficios económicos que estas soluciones les proporcionaban, mediante un análisis de costos/gastos en los que incurrían por utilizar soluciones de *software on-premise*.

1.3. Arquitectura *isolated tenancy*

Cuando las soluciones de *software* se empezaron a entregar como una solución *web*, se utilizó la arquitectura *isolated tenancy*. Dicha arquitectura requería que la solución se instalaba y se configuraba para cada uno de los clientes, que a su vez demandaba la configuración de un servidor, de la aplicación y demás pormenores requeridos. El proveedor del *software* era el responsable del mantenimiento de la solución, desde su instalación hasta su administración. Sin embargo, esto representaba algunas desventajas para el proveedor de la solución, que hacían inmanejable su modelo (Greer, M., 2009):

- Complejidad de administración: cada una de las soluciones instaladas/configuradas requería de una administración específica, ya que cada uno de los clientes tenía necesidades muy peculiares de su negocio.
- Alto costo de administración: debido a que cada cliente tenía necesidades específicas, la administración se tornaba muy compleja y por ende, los costos aumentaban: instalación, soporte, mantenimiento, actualizaciones, instalación de parches, etc. (Bloomberg, J., 2013)
- Poca escalabilidad: por cada cliente que requería de la solución, implicaba al proveedor realizar todo el proceso de instalación y configuración, lo cual hacía la solución poco escalable.

- Alto costo de infraestructura: cada cliente representaba para el proveedor, adquirir hardware adicional: almacenamiento, procesamiento, memoria y red.
- Alto riesgo: debido a la personalización de cada cliente, era necesario el lanzamiento de nuevas versiones para clientes específicos, que a su vez implicaba muchos riesgos al momento de su puesta en producción.

Este modelo traía beneficios principalmente para los clientes, quienes aprovechaban de la atención especial y personalizada para realizar más modificaciones a la solución de *software*. Sin embargo, conllevaba muchas desventajas y riesgos para el proveedor.

1.4. Soluciones existentes

En la actualidad, existen soluciones de *software* que permiten la gestión de procesos para ciertas industrias de *freelancing*, entre ellas la medicina y la abogacía. La mayoría de estas aplicaciones están diseñadas para cubrir necesidades específicas de sus industrias, por ejemplo, el seguimiento de la salud de un paciente, el seguimiento de un proceso legal de demanda, entre otras. Estas soluciones permiten muy baja adaptabilidad al negocio del cliente, lo que dificulta la entrega de una solución ampliamente utilizada por diferentes industrias. Algunas de las soluciones de *software* que se pueden encontrar en el mercado son:

- Clio: solución canadiense para la industria de abogacía, enfocada en las diferentes prácticas que se ejercen en la misma: ley criminal, ley de la familia, Gobierno, ley de inmigración, ley de bienes raíces, etc. Es una solución SaaS que permite la entrega inmediata de un *software* que cumple con las necesidades de un abogado, mediante una suscripción mensual. A pesar de ser una solución bastante completa, tiene un costo muy alto para el mercado latinoamericano, en donde los usuarios no están acostumbrados a pagar una mensualidad que llegue a impactar sus bolsillos. (www.clio.com, 2017)

- QuicDoc on cloud: solución estadounidense enfocada específicamente en la industria de la medicina, para la administración de los registros médicos de los pacientes. Es una solución en la nube con una suscripción mensual que ofrece diversas políticas de seguridad a nivel de infraestructura y de datos. (www.docutracinc.com, 2017)
- IBM Case Manager on Cloud: una solución robusta y flexible ofrecida por IBM, que se puede adaptar a diferentes industrias como: gobierno, seguros, banca, salud. A pesar de ser una solución en la nube, no permite la entrega inmediata de la misma, ya que es necesario contactar previamente al área de ventas para una cotización. (Zhu *et al.*, 2105)

A pesar de ser las soluciones más robustas de cada una de las industrias, son soluciones que no están al alcance de las pequeñas, medianas empresas de Latinoamérica por sus costos, y menos al alcance de *freelancers*, que aumentan día con día requiriendo de soluciones que les permitan entregar de forma más rápida sus servicios a sus clientes.

2. JUSTIFICACIÓN

La línea de investigación que persigue este proyecto se enfoca en los sistemas para impulsar el uso de *Cloud Computing* en los negocios, proponiendo el uso de soluciones *web* como un servicio para mejorar la gestión de las operaciones de los *freelancers*.

Actualmente existen diversas soluciones *web* que cubren parcialmente las necesidades de los *freelancers*; sin embargo, se enfocan en industrias específicas y tienen costos muy altos para los pequeños mercados. En esta situación, se propone una solución *web* como servicio, SaaS, que sea escalable, segura y que tenga costos accesibles, teniendo como principal objetivo cubrir las necesidades de los *freelancers*. Esto permitirá que los *freelancers*, puedan hacer uso de una solución que esté a su alcance en el aspecto económico, y que al mismo tiempo, les brinde la confianza de almacenar información confidencial y documentación importante para sus operaciones.

Por otro lado, mediante la prestación de una solución *web* como servicio, el proveedor debe procurar la creación de un modelo de negocio que le permita generar, en primera instancia, sostenibilidad y posteriormente una alta rentabilidad. Esto se logra mediante la combinación de una arquitectura de *software* escalable y tecnologías de punta, para poseer la capacidad de brindar el servicio a la mayor cantidad de clientes posibles y así generar un alto margen de ganancia.

3. ALCANCES

3.1. Perspectiva investigativa

El resultado de la investigación expondrá los recursos, procedimientos y componentes que deben ser considerados para la entrega de una solución tecnológica en la nube de forma automatizada, logrando una implementación inmediata y segura. De igual manera, expondrá el modelo de negocio a utilizar, que será determinante para la captación de clientes, según el mercado objetivo.

3.2. Perspectiva técnica

Desde un punto de vista técnico, el estudio presentará los diferentes aspectos tecnológicos que se deben de considerar para la implementación de una solución SaaS, las cuales por definición deben responder a una implementación inmediata para su satisfactoria entrega a los usuarios finales, siempre manteniendo estándares de seguridad que posibiliten la confidencialidad de la información.

Para este estudio, se tiene considerado utilizar las tecnologías de *Amazon Web Services* (AWS), ya que estas cuentan con características y servicios que facilitan la entrega de soluciones SaaS, permitiendo administrar procedimientos de recuperación y de automatización de escalamiento de recursos de computación, los cuales reducen las posibilidades de una caída en una solución SaaS.

Por lo tanto, se describen los aspectos técnicos que se cubrirán en el presente estudio.

- Procedimientos de aprovisionamiento de recursos: detallará todos los procedimientos técnicos necesarios para la asignación de recursos computacionales a un cliente de la solución SaaS. También se considerará el

escenario en el que se haya alcanzado un límite de los recursos computacionales y se tenga que proceder a un escalamiento horizontal para satisfacer la carga de trabajo, según la demanda requerida.

- Estándares de seguridad: identificará todos aquellos estándares de seguridad elementales para la entrega de una solución SaaS, por definición, los usuarios acceden a ella, mediante un acceso por internet, y esto a su vez, puede conllevar riesgos de fuga de información o de acceso no autorizado a la misma. Los estándares de seguridad se definirán a cuatro niveles: comunicación, aplicación, infraestructura de servidores y base de datos.

En cuanto al modelo de negocio, se presentarán indicadores que reflejen la rentabilidad de la solución y la ventaja competitiva, tomando como punto de partida los siguientes elementos:

- Pronósticos de los ingresos basados una posible cartera de clientes.
- Estimaciones de los costos de ventas que estarán en función de la demanda, de tal manera que mientras más clientes se tengan, mayor será el costo de ventas: servidores, almacenamiento, entre otros.
- Estimaciones de gastos con base a la demanda, y al costo de oportunidad que puede representar el incremento en el personal.

3.3. Resultados esperados

Una vez finalizado el estudio de los diversos aspectos técnicos considerados en la solución SaaS, se realizarán análisis estadísticos que permitirán medir la efectividad de los procedimientos utilizados en la solución para determinar si la entrega de la solución se ha realizado de forma exitosa cumpliendo con los estándares de seguridad requeridos.

De igual manera, se evaluará el modelo de negocio planteado, tanto desde un aspecto de rentabilidad que ayudará a determinar si su implementación genera una alta tasa de rentabilidad, y desde el aspecto de competitividad, que

ayudará a identificar factores clave que colaboren en la consolidación y ampliación de la cartera de clientes.

Siendo así, los resultados esperados deberán ser:

- Diseño de los procedimientos de aprovisionamiento de computación, memoria, *networking* y almacenamiento en disco que permitan la implementación inmediata y 100 % funcional de un *software* como servicio.
- Identificación de los componentes necesarios para la creación de una infraestructura en la nube que cumpla con estándares de seguridad.
- Definición del modelo de negocio a utilizar que capte la mayor cantidad de usuarios para crear una solución rentable.

4. MARCO TEÓRICO

En un escenario tradicional, se considera que el departamento de tecnología de una empresa, realice inversión de capital en tres áreas (Greer, M., 2009):

- *Software*: programas computacionales para la operación y análisis de información del negocio.
- Hardware: infraestructura utilizada para hospedar, comunicar, almacenar los programas computacionales.
- Personal: personas calificadas para el mantenimiento y administración del *software* y hardware de la empresa.

Adicional a estas tres áreas, se pueden generar gastos extras por una mala gestión de las anteriores, implicando mayor inversión en las áreas.

El área que representa mayor prioridad en el departamento de tecnología, es la de *software*, ya que es con la cual interactúan los usuarios finales para realizar las operaciones de la empresa. Por otro lado, el área de hardware y personal, son áreas de soporte al área de *software*, ya que permiten optimizar la entrega de las soluciones a los usuarios y brindan mantenimiento a las mismas soluciones.

A partir de las áreas mencionadas anteriormente, existen diferentes enfoques que ayudan a los departamentos de tecnología a adecuar sus servicios, según las capacidades de la empresa. Años atrás, cuando el uso de internet no era tan masivo, las empresas veían restringidas sus capacidades de comunicación y de uso de soluciones *web*. Por tal razón, las empresas que deseaban hacer uso de un *software* estaban obligadas a realizar una fuerte inversión en dos de las áreas mencionadas anteriormente: hardware, y por

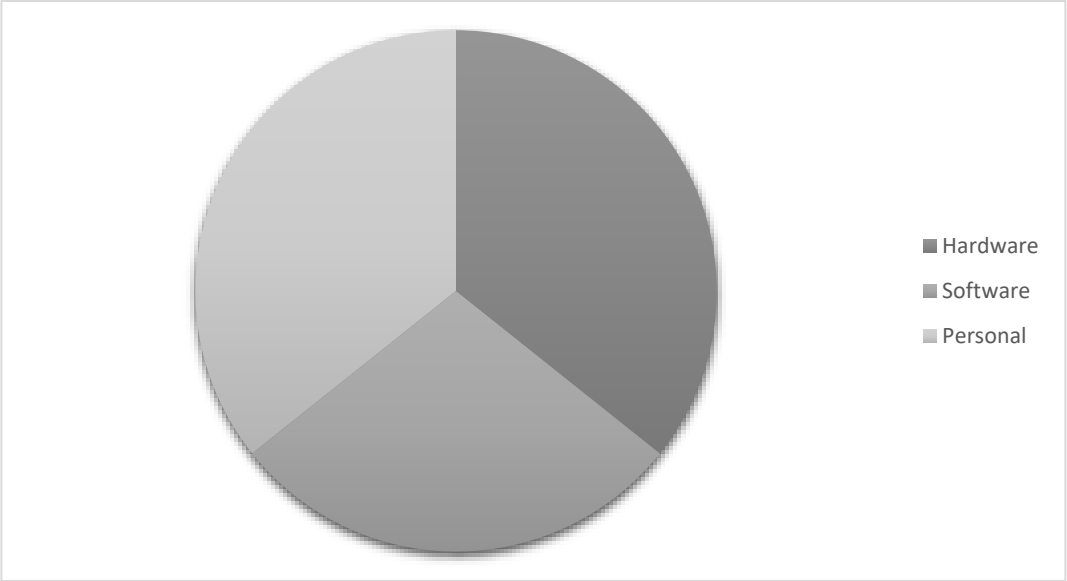
consiguiente, personal, dando como resultado un modelo *on-premise*. Sin embargo, hoy en día, dadas las capacidades del internet, ya no es necesaria una fuerte inversión en hardware para hacer uso de soluciones de *software*, ya que estas pueden estar alojadas por terceros proveedores, quienes brinden el servicio de uso del *software*, a lo cual se le denomina modelo SaaS.

Ambos modelos representan alternativas para la prestación de servicios de *software*, con variaciones de inversión en las áreas de tecnología, lo cual puede constituir una reducción a los costos de prestación del servicio. A pesar que cada modelo expone ventajas y/o desventajas al usuario, ambos modelos buscan mejorar ciertas necesidades: costos, seguridad, usabilidad, experiencia del usuario, entre otros.

4.1. Modelo *on-premise*

En el modelo *on-premise* como se ve en la Figura 1, la inversión en hardware y personal es mayor, debido a que requiere demasiada infraestructura y soporte para certificar que las soluciones de *software* están siendo entregadas de forma óptima a los usuarios, lo que permite que el trabajo no se vea afectado por fallas en la red o por un mal rendimiento de las aplicaciones. (Greer, M., 2009)

Figura 1. **Costo con sistemas *on-promise***



Fuente: Greer, M., 2009

En este modelo, la inversión en *software* consiste en la compra de licencias para el uso de las aplicaciones. En la mayoría de los casos, el adquirir el licenciamiento de *software* consiste en:

- Pago inicial por la compra del *software.*
- Uso del *software* durante un año.
- Uso de la garantía del *software*, para el reporte y corrección de fallas en la solución.

Sin embargo, al adquirir un licenciamiento el cliente no siempre tiene derecho a actualizaciones del *software* después de un año. En vez de ser un beneficio incluido por parte del proveedor, se comercializa con un pago anual adicional que da derecho a:

- Acceso a actualizaciones del *software.*
- Soporte por parte del proveedor.
- Acceso a bases de conocimiento.

- Acceso a *plugins* o *add-ons* de la aplicación.

Como resultado, la adquisición de una solución de *software on-premise* conlleva de dos inversiones: la inicial en donde se realiza la compra del *software* y una inversión anual que da acceso a futuras versiones, la cual puede equivaler a un 20 – 25 % de la inversión inicial.

4.2. Modelo SaaS

En un modelo de inversión en soluciones SaaS, se presenta una redistribución del presupuesto de los departamentos de tecnología, en donde el área de mayor inversión pasa a ser el área de *software*, reduciendo el costo de inversión en hardware y recurso humano.

Esto se debe principalmente a que un modelo de SaaS tiene diversas características que lo diferencian de las soluciones *on-premise*, las cuales son un punto de atracción e interés para los clientes. Entre las características de SaaS, se pueden encontrar:

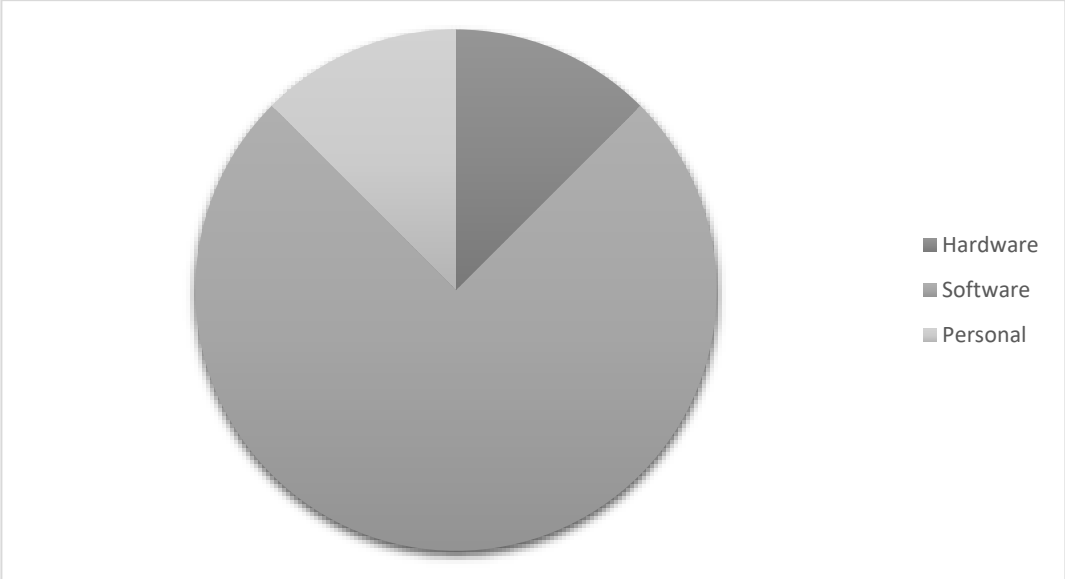
- No requiere de compra de licencias.
- Se paga únicamente por las funcionalidades utilizadas.
- Se adquiere el *software* mediante una suscripción.
- No requiere de hardware.
- Es accesible desde cualquier lugar.
- Es accesible en cualquier momento (Bouzid, A., 2015).
- No requiere de instalación de *software.*
- No requiere de mantenimiento y/o administración.
- Presenta mejoras constantes.
- Presenta nuevas funcionalidades constantemente.
- Ofrece soporte continuo.

Las anteriores características de un modelo SaaS, permiten que las empresas reduzcan sus costos de operación en el departamento de tecnología,

invirtiendo menos en hardware y personal, como consecuencia de una administración que está completamente atribuida al proveedor de la solución SaaS. Es importante resaltar que no requiere altos costos de inversión en hardware, porque no se basa en infraestructura física para hacer uso de las soluciones SaaS y tampoco para entregarla a todos los usuarios en las diferentes ubicaciones de la empresa.

El resultado es una redistribución en los costos por área, como se muestra en la siguiente gráfica. Ver Figura 2.

Figura 2. **Costo con soluciones SaaS**



Fuente: Greer, M., 2009.

4.2.1. Arquitectura de una solución SaaS

Una solución SaaS permite que muchos usuarios utilicen una misma aplicación para la realización de tareas u operaciones por internet. Los usuarios pagan una suscripción al proveedor, quien es el responsable de entregar una solución:

- Altamente disponible, asegurando que el usuario pueda tener acceso a la aplicación un 99.99 % del tiempo.
- Altamente utilizable, permitiendo a los usuarios utilizar la aplicación de forma sencilla y fácil, sin mayores conocimientos técnicos para el desempeño de sus operaciones.

Para esto, los proveedores deben poseer técnicas que habiliten la reutilización de una aplicación para todos los usuarios o clientes. De esta manera, los proveedores ponen en práctica el concepto de *muli-tenancy*, que se refiere a un principio en la arquitectura de *software* en la que una instancia del *software* es utilizada para atender a diversos clientes (*tenants*).

Dentro del concepto de *multi-tenancy*, existen varias arquitecturas y cada una de ellas presenta ventajas o desventajas, tanto para el cliente como para el proveedor. Algunas de las arquitecturas *multi-tenant* más conocidas se muestran en la Tabla III (Greer, M., 2009):

Tabla III. **Arquitecturas Multi-tenant**

Componente	Multi-tenant Simple	Multi-tenant Múltiple
Aplicación	Compartida	Compartida
Recursos	Compartidos	Compartidos
Separación de datos	Física	Lógica

Fuente: elaboración propia.

Existen otras arquitecturas *multi-tenant*, sin embargo, se evaluarán las dos arquitecturas listadas previamente, debido a que representan mayor beneficio para el proveedor en términos de costos.

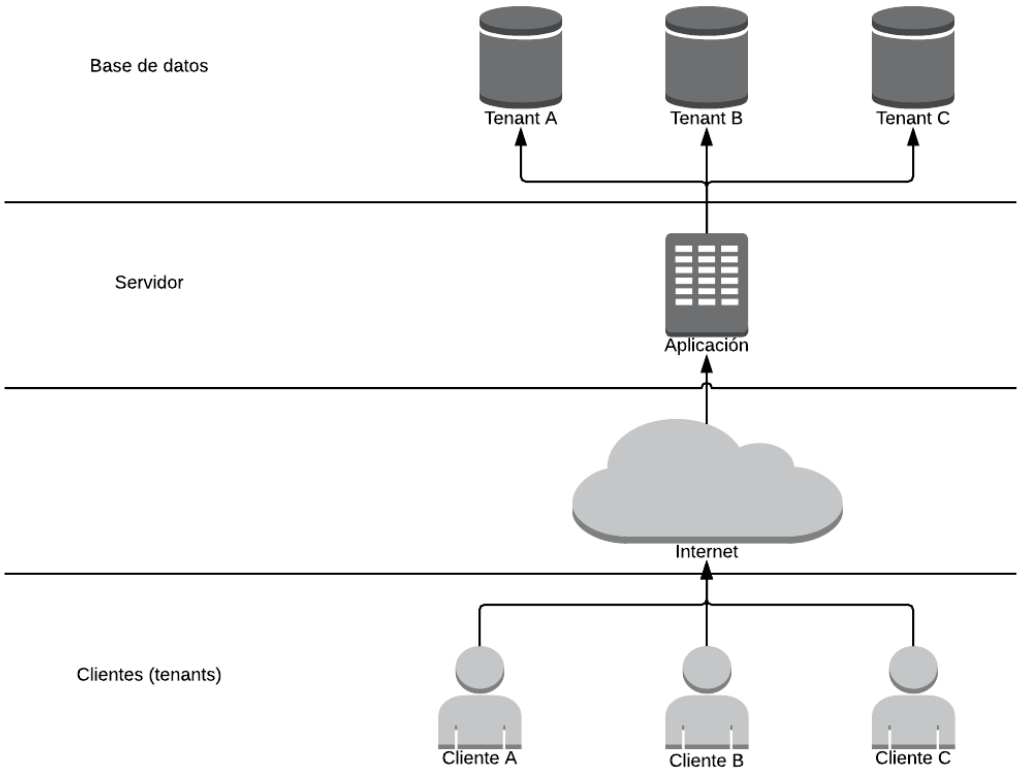
4.2.1.1. Arquitectura multi-tenant simple

En esta arquitectura, la aplicación y recursos de procesamiento son compartidos por todos los clientes; sin embargo, los recursos de almacenamiento utilizados en la aplicación, son únicos por cliente.

En otras palabras, los recursos de aplicación, incluyendo: procesador, memoria y espacio en disco para la aplicación, son compartidos por todos los clientes. Sin embargo, los recursos de almacenamiento para los datos del cliente, son únicos, de tal manera, que al momento de la instalación o configuración de la aplicación para el cliente, se reserva una base de datos dedicada especialmente para el cliente, permitiendo aislar por completo toda la información.

En el diagrama de la Figura 3, mostrado a continuación, se puede visualizar de mejor manera:

Figura 3. **Arquitectura *multi-tenant* simple**



Fuente: Greer, M., 2009.

En esta arquitectura de SaaS, se presentan las siguientes ventajas y desventajas. Ver Tabla IV.

Tabla IV. **Ventajas y desventajas de arquitectura multi-tenant simple**

Ventajas	Desventajas
• Aislamiento de información de cada cliente • Mejor tiempo de respuesta de acceso a la información • Facilidad al realizar respaldos de bases de datos para clientes específicos • Facilidad al restaurar respaldos de bases de datos para clientes específicos	• Mayor costo de almacenamiento • Mayor complejidad en el versionamiento de base de datos • Mayor complejidad en las tareas de mantenimiento de base de datos: respaldos generales y optimización

Fuente: elaboración propia.

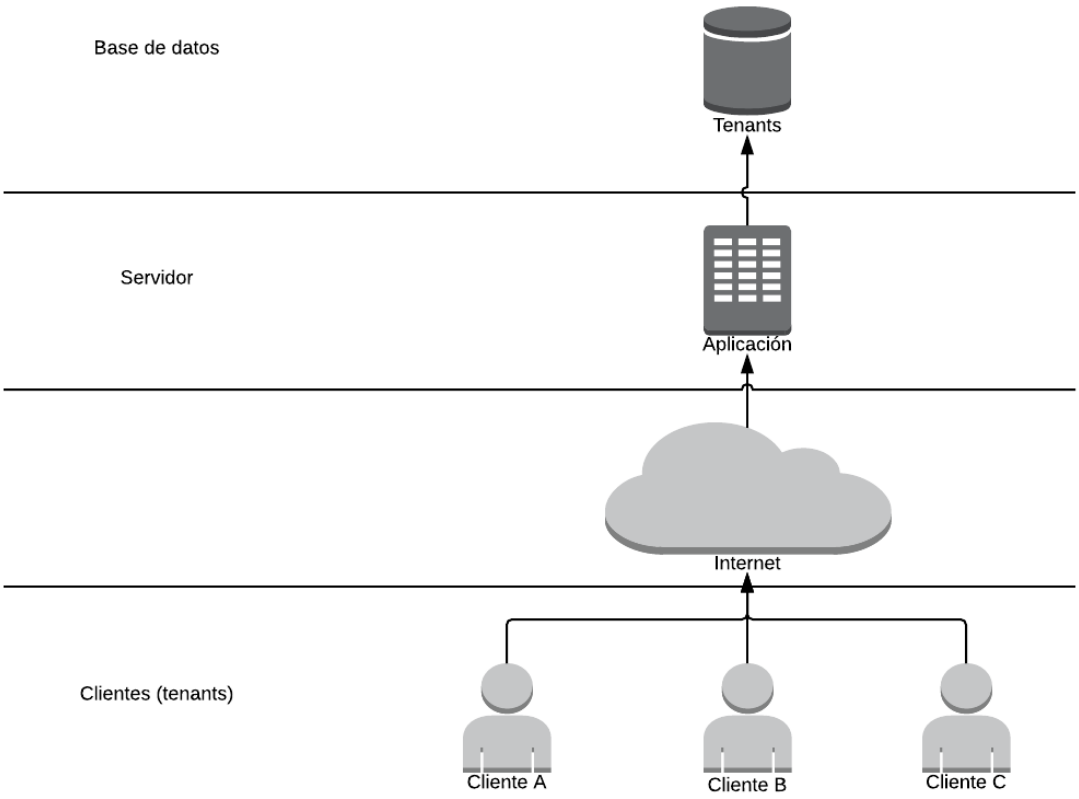
4.2.1.2. Arquitectura multi-tenant múltiple

La arquitectura *multi-tenant* múltiple representa menor costo para el proveedor, debido a que se comparten todos los recursos de los clientes: procesador, memoria, espacio en disco y almacenamiento con base de datos.

En este caso, la base de datos es compartida, realizando la separación de datos a nivel lógico, mediante la aplicación, sirviendo al cliente únicamente la información que corresponde al cliente.

En la Figura 4 se puede visualizar la arquitectura *multi-tenant* múltiple:

Figura 4. **Arquitectura *multi-tenant* múltiple**



Fuente: Zhu, Zhang, Li, 2012.

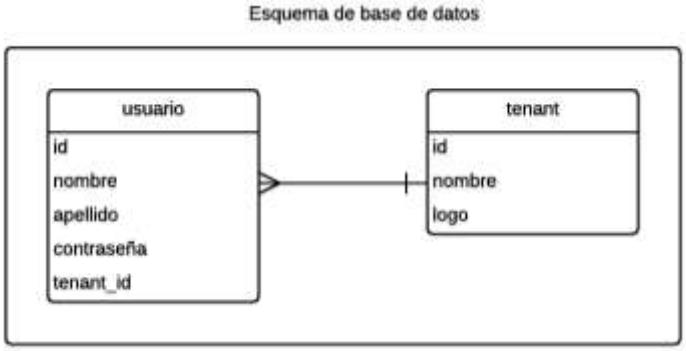
En este caso, toda la información de los clientes reside en una misma base de datos, permitiendo reducir los costos de almacenamiento; sin embargo, esto requiere que a nivel de aplicación se realicen validaciones adicionales que cumplan con el objetivo de mostrar al cliente únicamente su información. (Zhu *et al.*, 2012)

Estas validaciones se realizan a nivel lógico de la base de datos, haciendo uso de una tabla maestra que habilite la gestión de los clientes, la cual será utilizada para filtrar las peticiones en el resto de tablas.

El diseño de base de datos requiere de las siguientes características, ver Figura 5:

- Una tabla maestra de clientes que contenga cada uno de los clientes existentes de la solución con su respectivo identificador.
- Por cada tabla de aplicación, se utiliza una columna adicional que identifica al cliente al cual pertenece el registro, haciendo referencia a la tabla maestra de clientes.

Figura 5. **Base de datos de arquitectura *multi-tenant* múltiple**



Fuente: Zhu *et al.*, 2012.

En esta arquitectura de SaaS, se presentan las siguientes ventajas y desventajas. Ver Tabla V.

Tabla V. **Ventajas y desventajas de arquitectura multi-tenant múltiple**

Ventajas	Desventajas
• Menor costo de almacenamiento • Facilidad en el versionamiento de base de datos, ya que solo es requerido ejecutar las modificaciones en una base • Facilidad al realizar tareas de mantenimiento de base de	• No se tiene aislamiento de la información • Tiempo de respuesta de acceso a la información afectado, debido a la cantidad de información de los diversos clientes

datos: respaldos generales y optimización	• Mayor complejidad al realizar respaldos para clientes específicos • Mayor complejidad al restaurar respaldos de bases de datos para clientes específicos

Fuente: elaboración propia.

4.2.2. Seguridad en soluciones SaaS

La información de cualquier institución, sea pública o privada, de lucro o no de lucro, local o internacional, es un activo valioso que debe ser resguardado con el objetivo de impedir que se filtre información sensible de la organización, por ejemplo: ventas, clientes, proveedores, costos, tiempos de entrega, tiempos de producción, etc. Cuando la información es vulnerada, existen riesgos de manipulación, destrucción, desventaja de estrategia competitiva, entre una serie de circunstancias que afecten la integridad de la empresa.

En toda solución de *software,* la seguridad desempeña un papel de suma importancia. Mientras más segura sea la solución, más confiable será el proveedor, logrando credibilidad en el mercado para el uso de sus soluciones. Esta situación, es más evidente en las soluciones SaaS, ya que se presta un servicio a una cantidad de clientes que aumenta de durante el tiempo.

Es imprescindible que el proveedor del SaaS, cumpla con estándares de seguridad sobre su solución, para certificar que su servicio cuenta con canales de comunicación confiables, bloqueos a amenazas de terceros, privacidad en los datos empresariales, etc. Teniendo como prioridad, minimizar o reducir las probabilidades de ataques sobre la integridad de la información.

Para lograr una mayor confiabilidad, las aplicaciones deben ser alojadas en infraestructuras seguras, ya sean implementadas por el mismo proveedor o por proveedores terceros, tales como las infraestructuras facilitadas por proveedores

de IaaS (*Infrastructure as a Service*) como: *Amazon Web Services*, *Google Cloud* o *Microsoft Azure*, entre otros. Sea cual sea la infraestructura a utilizar, el proveedor de la solución SaaS debe garantizar en todo momento, la privacidad y la veracidad integridad de la información. (Krause, M., 2011)

Para esto, se deben considerar ciertos aspectos técnicos que permitan mantener la información de los clientes a salvo y libre de cualquier amenaza. A continuación, se describen las diversas capas de la seguridad en una solución SaaS, que deben tener mayor relevancia.

4.2.2.1. Aplicación

La aplicación es la puerta de entrada para el uso de la solución SaaS, y es la que está al alcance de los usuarios para su uso correspondiente. A pesar de que la aplicación pudiese contar con métodos de autenticación, cualquier persona puede llegar a acceder a ella mediante su suscripción al servicio por el pago respectivo. Muchos de los usuarios pueden considerarse usuarios nobles, que no buscan realizar ningún uso deshonesto sobre la aplicación; sin embargo, existen usuarios maliciosos, que únicamente crean amenazas o buscan fallas sobre la aplicación para aprovecharse de las mismas, y sacar ventaja de ellas.

Por esta razón, la seguridad a nivel de aplicación, definitivamente puede representar el aspecto con mayor prioridad en cuanto a la seguridad de la solución, dado que por ella pueden ingresar infinidad de ataques en contra de la información. Sin necesidad de forzar o violentar las políticas de seguridad de la aplicación, pueden ocurrir diferentes ataques provenientes de ella que ocasionen alguna alteración a la información, o incluso, el acceso no autorizado a información sensible. Es importante que a nivel de aplicación, se realicen continuamente estudios que permitan identificar agujeros en la seguridad de la aplicación, o bien, identificar nuevas amenazas en el entorno que pudiesen afectar la estabilidad de la aplicación.

Los aspectos que comúnmente se consideran para tener una aplicación segura pueden ser:

- Uso de fuertes algoritmos o métodos de autenticación.
- Control de acceso sobre las funciones correspondientes.
- Control de acceso a la información, según el rol del usuario.
- Métodos para obstaculizar las técnicas de inyección SQL (Shema, M., 2012).
- Uso adecuado de las variables de sesión para contrarrestar el secuestro de sesiones.
- Bloqueo de cuentas después de varios intentos de autenticación fallidos.
- Métodos para evitar Cross-site scripting (Lepofsky, R., 2014).

4.2.2.2. Infraestructura

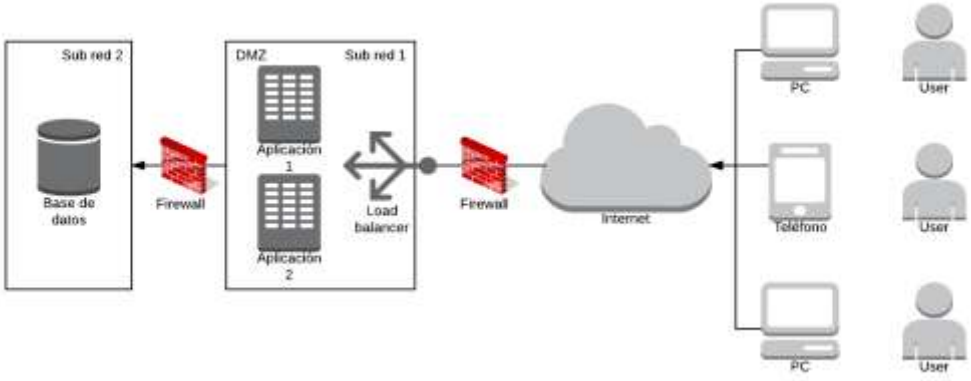
Una aplicación requiere de diversos componentes para su correcta entrega al usuario final, dentro de los cuales se pueden considerar componentes de comunicación y de alojamiento, que permiten la transmisión de la información desde la ubicación del servidor de la aplicación, hasta la ubicación del usuario. De igual manera, la aplicación requiere que su alojamiento sea en un lugar seguro libre de ataques que puedan perjudicar la integridad de la información del usuario, mediante el uso de hardware que bloquee estas amenazas.

Dicha comunicación, desde y hacia la aplicación se realiza mediante la red, ya sea del lado interno de la infraestructura del proveedor de la solución SaaS, o del lado externo del proveedor, entiéndase este último como el internet llegando hasta la red interna del usuario de la aplicación. Si bien es cierto, el proveedor no es responsable de la red interna del usuario, es responsable que toda la comunicación entre ambas partes (proveedor-usuario y usuario-proveedor) se realice mediante los canales más seguros posibles. (Mather, Kumaraswamy, Tipton, 2009) Así mismo, se debe asegurar que los usuarios únicamente tengan acceso a los puertos correspondientes para el uso de la aplicación.

Adicionalmente, para asegurar que la comunicación entre el usuario y la solución SaaS cumple en todo momento con los procedimientos más confiables de transmisión, es recomendable realizar un correcto diseño de infraestructura y de la red interna, que cuente con las más estrictas políticas de acceso a los servidores y bases de datos. (Migga, J., 2015)

A continuación, en la Figura 6, se muestra un diagrama de infraestructura para crear zonas de aislamiento de cada uno de los componentes que alojan y comunican la aplicación.

Figura 6. **Diagrama de infraestructura para soluciones *web* públicas**



Fuente: docs.aws.amazon.com/AmazonVPC, 2017.

Con base al anterior diagrama, se pueden identificar varios puntos clave para la seguridad a nivel de infraestructura y de red de una solución *web* de *software*:

- Protocolos de comunicación seguros entre el usuario y la aplicación (HTTPS).
- Control de acceso a las subredes mediante el uso de ACL's (*Access Control List*) (Mather *et al.*, 2009).
- Uso de sub redes para aislar los componentes de la solución.
- Configuración de una DMZ (*Demilitarized Zone*) para los servidores que son accedidos desde redes externas o internet.

- Uso de firewall para el bloqueo de puertos (Migga, J., 2015).
- Aislamiento de servidores de bases de datos en subredes adicionales.

4.2.2.3. Almacenamiento

Como se ha mencionado en secciones previas, la información es el principal activo de toda organización, y el objetivo de todo esquema de seguridad debe ser proteger su integridad y su privacidad. En el ámbito de tecnología, los proveedores de soluciones de *software* deben ser extremadamente cuidadosos con la administración de la información de sus clientes, por ende, deben llevar a cabo la implementación de políticas de seguridad que brinden una solución confiable a los usuarios.

Hoy en día, en soluciones SaaS, el almacenamiento de información no solo responde a necesidades de bases de datos, sino también a necesidades de almacenamiento de documentos digitalizados, archivos multimedia, documentos de texto, hojas de cálculo, entre otros. De esta manera, la seguridad a nivel de almacenamiento se ha diversificado, ya que ha pasado de ser de un solo plano, en donde se almacenaba toda la información en una base de datos, a almacenar diferentes tipos, formatos y estructuras de información. Algunos almacenamientos pueden ser:

- Bases de datos relacionales
    - Oracle
    - SQL Server
    - MySQL
    - PostgreSQL
- Bases de datos NoSQL
    - MongoDB
    - SimpleDB
- Data warehouse/BigData
    - Hadoop

    - Redshift
    - Sybase IQ
    - SAP Hana
- Multimedia
    - Google Drive
    - Dropbox
    - OneDrive
    - MEGA
    - AWS S3

Independientemente del tipo de almacenamiento, continúa siendo información muy importante para el cliente, y debe ser resguardada en todo momento por el proveedor de la solución SaaS para asegurar su disponibilidad.

Para esto, se deben considerar ciertos lineamientos que permitan certificar que la información proporcionada por el cliente, está siendo almacenada tal y como fue recibida; y que la información que se está entregando al cliente, es tal y como se recibió, sin sufrir de alguna alteración a la misma. (Mujawar, Sutagundar, Ragha, 2017). Algunos de los lineamientos son:

- Implementación de algoritmos de encriptación de la información durante la transmisión entre el proveedor y el cliente, y viceversa (SSL, HTTPS) (Rao, 2015).
- Encriptación de la información a nivel de almacenamiento: base de datos y dispositivos: disco duro, cintas, etc.
- Replicación de la información para asegurar su disponibilidad.
- Respaldos de la información (Rao, 2015).
- Registro del historial de los datos para permitir identificar quién y cuándo ha accedido a ellos (Mujawar *et al.,* 2017).

4.2.3. Modelos de negocio SaaS

El modelo de negocio de una solución SaaS puede llegar a ser determinante en la salud financiera de la empresa, y más aún, en el crecimiento de la misma, ya que influye en factores clave como: adquisición de clientes, fidelización de clientes, margen de ganancia, entre otros. Para una mejor identificación del modelo de negocio, es importante realizar un análisis sobre el mercado objetivo, se puedan encontrar las necesidades y los problemas por resolver de los posibles clientes, permitiendo brindarles una solución y un modelo de precios más accesible.

El modelo seleccionado ayudará a identificar el esfuerzo requerido para lograr la conversión de un posible cliente a un cliente, y posteriormente, identificar el tiempo en el que se logra recuperar la inversión realizada durante el proceso de adquisición mediante el pago de la suscripción mensual del cliente. El objetivo del modelo de negocio recae en lograr identificar y optimizar el flujo de efectivo del proveedor SaaS, y así determinar su viabilidad.

Las soluciones SaaS regularmente están basadas en modelos de suscripción, sin embargo, dentro de este modelo, existen variantes que pueden representar diferentes ventajas para los clientes, dependiendo del servicio que sea entregado. (Prabowo, Janssen, Barjis, 2012)

4.2.3.1. Tipos de modelos de negocio

Basado en suscripción por usuario

Este modelo requiere de una suscripción mensual o anual, que le da derecho de uso al cliente de la aplicación. En este sentido, el cliente puede utilizar la aplicación durante el tiempo que pague la suscripción, teniendo acceso a actualizaciones y soporte del *software*. Sin embargo, paga de acuerdo a la cantidad de usuarios que se tenga derecho.

Por ejemplo: Zendesk, Salesforce.

Basado en transacciones

Este modelo requiere de un pago mensual con base a operaciones realizadas en el *software*. A mayor uso de la aplicación, mayor será el pago de la suscripción.

Por ejemplo: Amazon AWS, Google Cloud.

Basado en suscripción fija

Este modelo requiere de un pago fijo mensual o anual, que le da derecho de uso a la aplicación realizando todas las operaciones deseadas. Sin embargo, se tiene un límite en la cantidad de usuarios.

Por ejemplo: Netflix, Spotify.

4.2.3.2. Comparativa de modelos SaaS

Existen otros modelos SaaS, pero los más conocidos o utilizados, son los descritos previamente, ya que representan la mayoría de ventajas que se pueden brindar a los clientes.

Es importante notar que el modelo de negocio, está en función del servicio brindado. Por ejemplo, Zendesk es un *software* SaaS, que brinda el servicio de acceso a una aplicación CRM. Un CRM por lo regular, es utilizado por diversos departamentos en una empresa, ya que puede formar parte de la estrategia en el equipo de ventas para el seguimiento de los clientes, o bien, puede ser utilizado como herramienta de soporte en el área de servicio al cliente. De esta manera, Zendesk ofrece sus diversos servicios como se muestra en la Tabla VI, con base a un modelo de suscripción por usuario. En el que cobra de forma mensual o anual, el acceso al servicio de forma ilimitada.

Tabla VI. **Servicios que ofrece Zendesk**

Funcionalidad	Suscripción por usuario	Transacciones	Suscripción fija
Actualizaciones	Sí	Sí	Sí
Soporte	Sí	Sí	Sí
Usuarios	Limitados	Ilimitados	Limitados
Usuarios concurrentes	No aplica	Ilimitados	No permitido
Transacciones	Ilimitadas	Limitadas	Ilimitadas

Fuente: elaboración propia.

5. PRESENTACIÓN DE RESULTADOS

5.1. Arquitectura de solución

5.1.1. Componentes principales

La solución propuesta en el presente documento considera diferentes escenarios que buscan la creación de una aplicación robusta, escalable, altamente disponible y segura, lo cual debe involucrar diferentes componentes.

A lo largo del documento, se detallarán las mínimas configuraciones correspondientes que forman parte de las áreas técnicas cubiertas en el estudio:

- Procedimientos de aprovisionamiento de recursos y
- Cumplimiento de estándares de seguridad

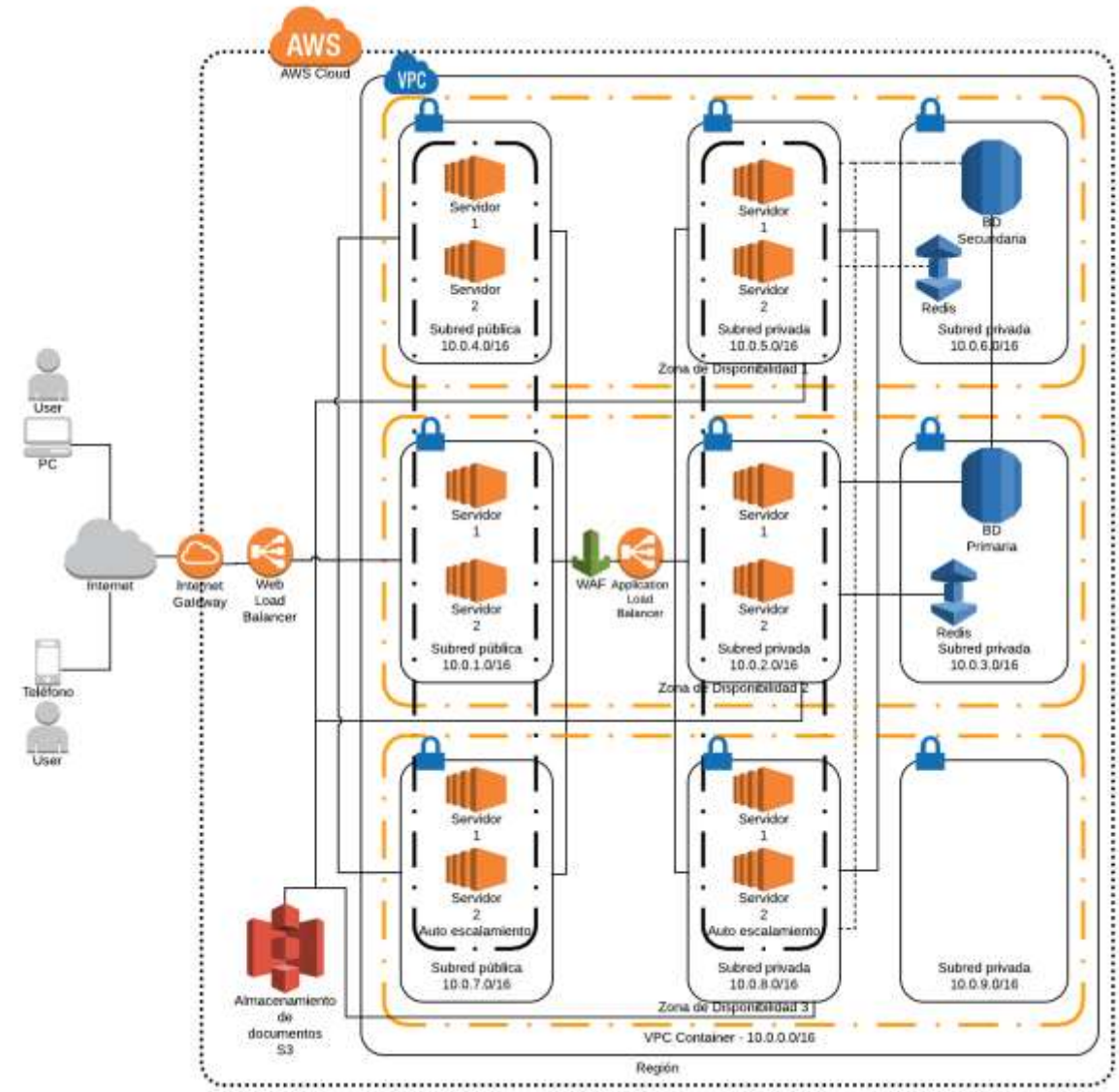
Para la cobertura de estas áreas se hacen uso de los siguientes componentes en la infraestructura:

- Región
- Red privada virtual
- Grupo de red
- Subredes
- Zonas de disponibilidad
- Servidores
- Internet gateway
- Balanceadores de carga
- *Web application firewall*
- Autoescalamiento
- Base de datos relacional
- Redis (*Amazon ElastiCache*)

- Almacenamiento de documentos (*Amazon S3*)

Estos componentes pertenecen al portafolio de servicios en la nube, brindados por *Amazon Web Services,* proveedor de *Cloud Computing.* En la Figura 7, se puede observar la interacción de los componentes mencionados anteriormente y en secciones posteriores, se detallará la funcionalidad de cada uno de ellos.

Figura 7. **Arquitectura de solución**



Fuente: elaboración propia.

5.1.2. Componentes clave adicionales

Dentro de la arquitectura propuesta en la anterior sección se despliegan los componentes principales para la entrega de la solución, en donde se consideran: servidores, balanceadores de carga, subredes, bases de datos, entre otros. Sin embargo, vale la pena señalar que existen recursos adicionales que deben ser

configurados para una correcta administración de la arquitectura, pero que no forman parte de los recursos principales de la solución.

A continuación, se detallan estos recursos y sus configuraciones básicas para su correcto uso.

5.1.2.1. NAT Gateway

En *Amazon Web Services* existe el término *NAT Gateway,* que corresponde a un recurso que se encarga de habilitar el acceso a internet desde subredes privadas sin permitir el acceso desde internet a los servidores que se encuentren alojados en dichas subredes, actuando como un mecanismo de seguridad, para impedir el acceso no deseado desde internet a servidores críticos de la solución.

El propósito del *NAT Gateway* es posibilitar el acceso a descargas de internet o sincronización de repositorios utilizados, para mantener actualizados los paquetes/librerías/*software* que se tengan instalados en los servidores. De esta manera se lograrían mantener los servidores con las versiones más recientes y seguras de los diferentes aplicativos, entregando una solución más robusta y confiable para los usuarios finales.

La configuración necesaria para el *NAT Gateway* corresponde a la asociación de una subred y de una IP elástica. En el caso de la subred, debe asociarse a la subred pública que forma parte de la arquitectura, y en cuanto a la IP Elástica, se debe crear una nueva si no se tiene una disponible. Esta configuración quedaría, según se muestra en la Figura 8.

Figura 8. **Configuración de de NAT Gateway**

Fuente: elaboración propia.

5.1.3. Recomendaciones de arquitectura

AWS EFS, la cual es una solución de sistemas de archivos que puede ser utilizada por diversos servidores de forma simultánea desde diferentes zonas de disponibilidad. Al mismo tiempo, es una solución completamente administrada por *Amazon Web Services,* lo cual representa otros beneficios en cuanto a la administración de los servidores.

Bastion Server, para que la configuración sea más fácil de mantener, también se recomienda el uso de una IP elástica para el *Bastion Server,* para su correspondiente configuración en los *Security groups* mencionados en la sección de Configuración de firewall. Adicionalmente, solo se configurarían los puertos necesarios para realizar las tareas de actualización, en este caso, se recomienda habilitar únicamente el puerto 22, del servicio SSH.

Se recomienda que a nivel del *Web Application Firewal*, se ejecute la creación de reglas, condiciones y acciones necesarias que monitoreen el puerto 80 y 443, para tener un registro de las diferentes peticiones realizadas y así proceder al bloqueo de las IP correspondientes. De esta manera, se lograría contrarrestar cualquier ataque de *Distributed Denial of Service* (DDoS) y bloquear cualquier amenaza al servicio para brindar una alta disponibilidad del mismo.

5.2. Procedimiento de aprovisionamiento de recursos

5.2.1. Arquitectura *multi-tenant* con uso de subdominio

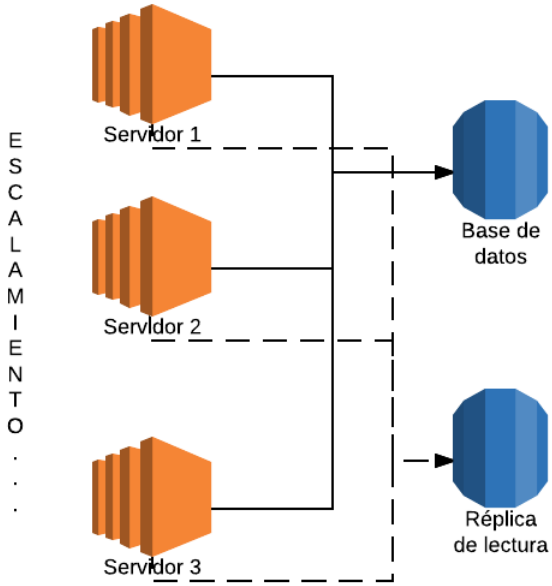
Toda aplicación *Software as a Service* tiene como propósito las economías de escala para incrementar su margen bruto. Para esto se deben utilizar ciertas técnicas que permitan escalar la solución de forma fácil, y así proveer el servicio a gran cantidad de clientes dando un único mantenimiento para todos ellos.

La escalabilidad en una solución de tipo *Software as a Service*, se logra utilizando una arquitectura *multi-tenant*. Al igual como se ha discutido en secciones anteriores, para este tipo de arquitectura existen diversas opciones; sin embargo, en el presente documento se tomará como base la arquitectura *multi-tenant* múltiple, la cual se ajusta de mejor manera a la solución que se está proponiendo.

La arquitectura *multi-tenant* múltiple debe cumplir con los siguientes requerimientos:

- Base de datos
    - Una base de datos que contenga la información de todos los clientes: usuarios, permisos, roles, facturas, etc. En este caso, se recomienda una base de datos RDS (Amazon Relational Database Service), la cual posibilita una fácil configuración, operación y escalabilidad de cualquier base de datos relacional. El propósito de una base de datos de este tipo, es que permite a una aplicación escalar de forma horizontal, es decir, agregar más pequeños servidores a la solución, sin necesidad que la aplicación se tenga que configurar nuevamente para apuntar hacia diferentes servidores de base de datos. Adicionalmente, que una base de datos RDS, tiene otras características fundamentales para lograr alta disponibilidad y seguridad en los datos, por ejemplo: replicación, backup y recuperación automatizada. La arquitectura de base de datos propuesta, se puede observar en la Figura 9.

Figura 9. **Arquitectura de base de datos**



Fuente: elaboración propia.

- Una tabla que almacene cada uno de los clientes y tenga un identificador único relacionado: *tenant.* Su estructura se puede observar en la Tabla VII.

Tabla VII. **Estructura de tabla de tenants**

Tenant		
Id	**subdomain**	**Name**
1	bufeteaya	Bufete Aguirre y Asoc.
2	medfmartinez	Médico Fernando Martinez

Fuente: elaboración propia.

- Un diseño en la base de datos en el que se agregue una columna *tenant_id* a cada una de las tablas del sistema.
- Cada registro insertado en la base de datos debe estar asociado a una cuenta, por lo tanto, dicho registro deberá tener en la columna

tenant_id el identificador asociado a la cuenta que se encuentra en la tabla *tenant.* Esta estructura se puede observar en la Tabla VIII.

Tabla VIII. **Estructura de tablas de aplicación**

User			
id	**Name**	**lastname**	**tenant_id**
1	Luis	Aguirre	1
2	Antonio	Dominguez	1
3	Fernando	Martinez	2
4	Sara	Lopez	2

Fuente: elaboración propia.

- Infraestructura
    - Se deberá contar con un dominio para albergar a cada uno de los clientes a quienes se les prestará el servicio. En la presente solución, se sugiere que cada cliente que desee crear una cuenta en el *software*, sea capaz de seleccionar un nombre para ella, y que este nombre sea utilizado como subdominio para que lo identifique de manera única. De tal manera que si el cliente desea ingresar a su aplicación, debe hacer uso de la siguiente URL: *https://nombre.aplicacion.com*

En donde la URL contiene los elementos listados en la Tabla IX:

Tabla IX. **Elementos de URL de clientes**

Elemento	Descripción
https	Protocolo sobre el cual se entregará la aplicación al usuario final. A diferencia del protocolo http, el protocolo https cifra la comunicación entre el servidor y el usuario, asegurando la confidencialidad de la información que se está transmitiendo.

Nombre	Nombre de la cuenta que el cliente haya seleccionado al momento de la creación/suscripción al servicio. Este nombre deberá ser único en todo el catálogo de clientes (tabla *tenant*), ya que se utilizará como subdominio para que el usuario pueda ingresar a utilizar la aplicación.
aplicación	Dominio sobre el cual se alojará la aplicación. Será el punto de partida para dar a conocer la aplicación, y sobre este mismo dominio se tendrá un sitio *web* que detalle los beneficios y funcionalidades de la aplicación con el objetivo de captar más cliente. Desde este mismo dominio, se le permitirá al usuario suscribirse creando su propia cuenta.
Com	Dominio de primer nivel sobre el cual se alojará la aplicación. Se utiliza .com debido a que la aplicación tiene un propósito comercial.

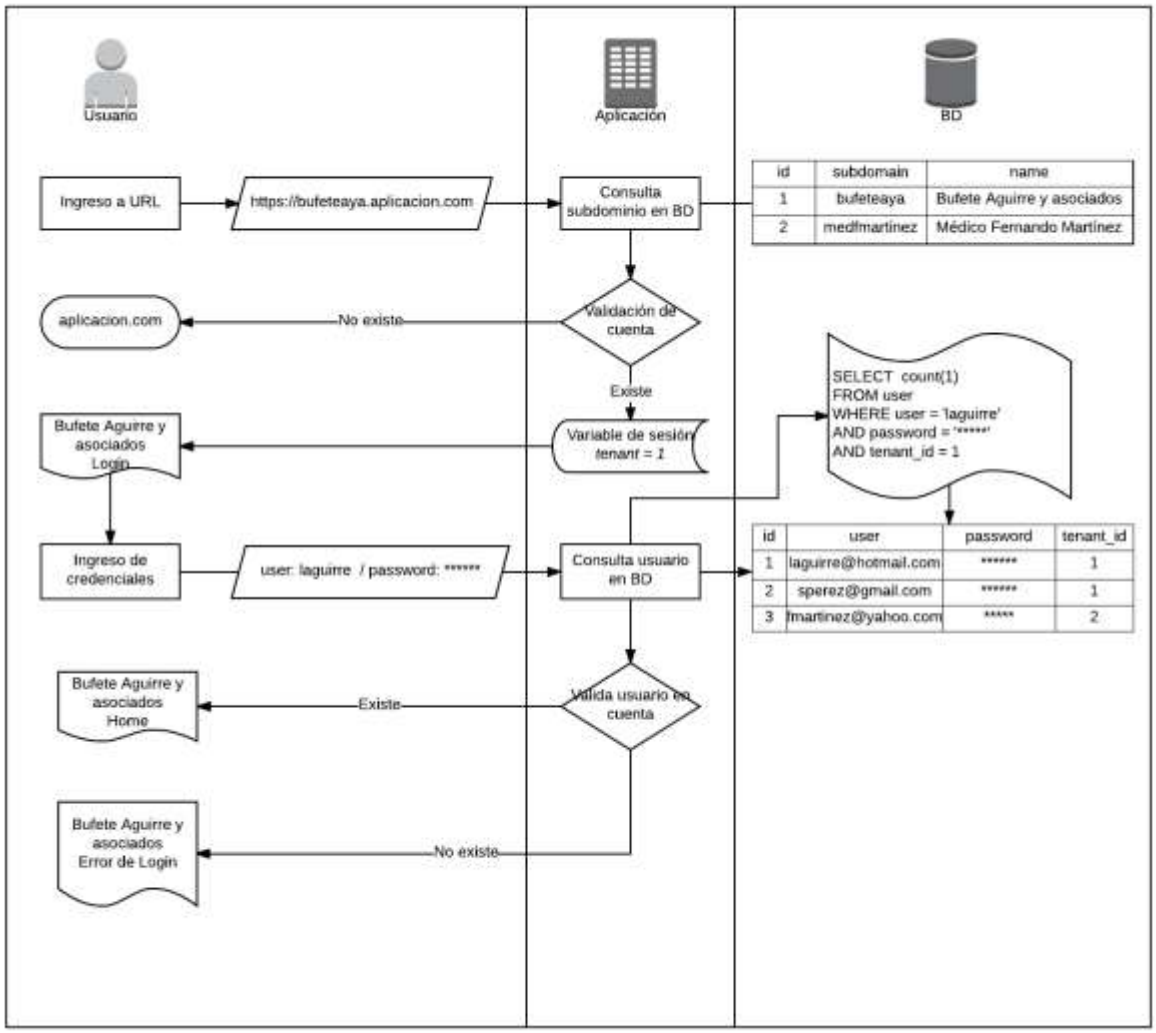
Fuente: elaboración propia.

- Aplicación
    - Una vez definida la URL del cliente, esta se utilizará para identificar a qué *id* de la tabla *tenant* corresponde su información. Por lo tanto, al momento que el usuario ingrese a la URL, la aplicación se encargará de validar la existencia del subdominio en la base de datos en la tabla *tenant.* Si este existe, guarda en una variable de sesión el *id* correspondiente, de lo contrario, lo redirecciona a la página de la aplicación.
    - Para cada consulta realizada por el usuario final en la aplicación, esta deberá añadir una condición a la cláusula WHERE en el SQL generado, que permita filtrar la información a devolver, de tal manera que el usuario únicamente sea capaz de tener acceso a la información de su cuenta.

- La aplicación será instalada y configurada en cada servidor que forme parte de la infraestructura de la solución. Este proceso se definirá en secciones posteriores.

El flujo que debe seguir la aplicación para una correcta gestión de la información del cliente, se puede observar en el diagrama de la Figura 10.

Figura 10. **Creación de cuenta**



Fuente: elaboración propia.

5.2.2. Evaluación de disponibilidad de subdominio (cliente)

Una solución *Software as a service* debe permitir la creación de una cuenta con su respectiva configuración de forma automática directamente desde el sitio. Esto hará que las probabilidades de un usuario que ingresa al sitio de la aplicación, en búsqueda de sus funcionalidades o de información, se convierta en un cliente aumenten de forma potencial, ya que el usuario tendrá al alcance de sus manos el poder crear y obtener una “aplicación” que cubra sus necesidades en pocos pasos.

Para lograr esto, se debe proporcionar directamente desde la página principal del sitio una opción que le brinde al usuario la posibilidad de crear su cuenta. Esta opción básicamente será un formulario que deberá de incluir la menor cantidad de campos posibles, que el proceso de creación de cuenta sea fácil y rápido. Los campos que debe contener dicho formulario deberán ser:

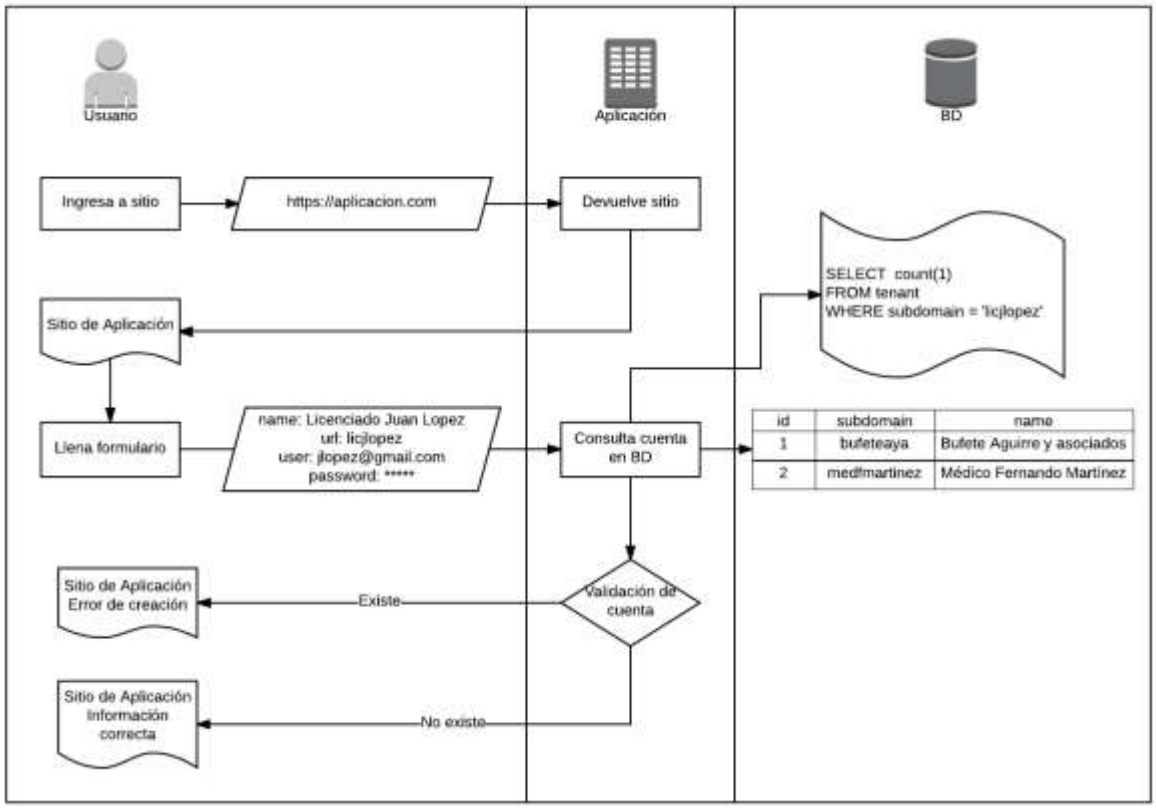
- Nombre.
- URL de cuenta (a utilizar como subdominio).
- Correo electrónico del usuario que crea la cuenta, futuro propietario de la misma.
- Contraseña del usuario que crea la cuenta.

Tal y como se indica anteriormente, el usuario que crea la cuenta se convierte en propietario de la misma, de tal manera que, ese usuario tendrá permisos de súper administrador para realizar cualquier modificación sobre la configuración de la cuenta. Sin embargo, se debe permitir que el correo electrónico de dicho usuario se pueda modificar para cubrir cualquier cambio que se haga en la organización.

Una vez ingresados los datos requeridos en el formulario, la aplicación deberá llevar a cabo un pequeño proceso que sea capaz de identificar la disponibilidad de la URL. Esto se hará mediante un procedimiento que consulte la tabla *tenant* de la base de datos, la existencia de dicho subdominio. Si el subdominio ya existe en la tabla mencionada, se le deberá desplegar el error correspondiente al usuario indicando que la URL ya fue tomada y que es necesario que intente con otra URL. De lo contrario, se continuará con el proceso de creación de la cuenta.

El proceso a realizar se puede observar en la Figura 11.

Figura 11. **Evaluación de disponibilidad de subdominio**



Fuente: elaboración propia.

5.2.3. Asignación de subdominio (cliente)

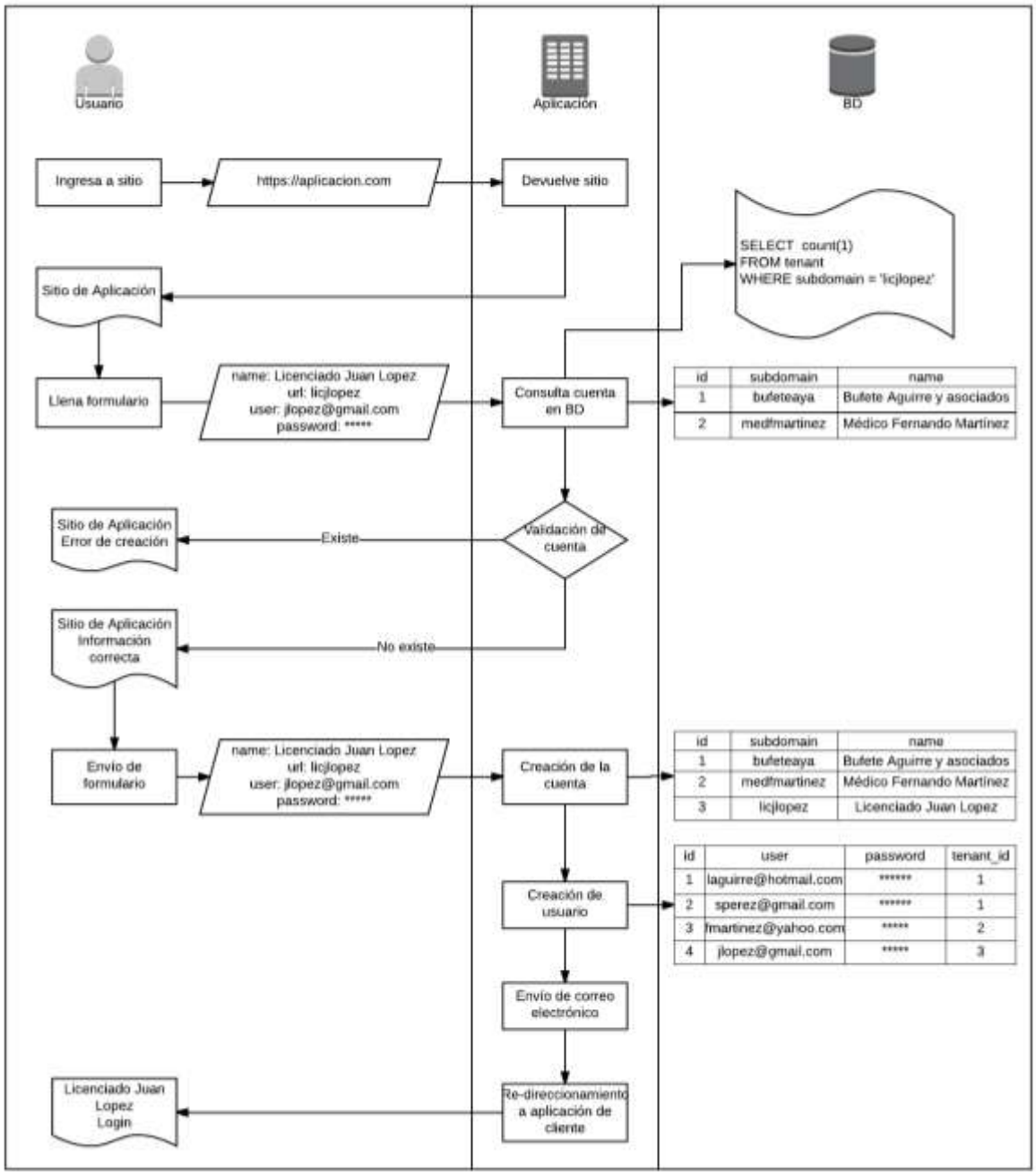
El proceso de validación de disponibilidad de subdominio es el primer paso para la creación de la cuenta. Una vez se haya obtenido un resultado exitoso de la disponibilidad, se debe proceder con la asignación de la URL como subdominio para que el cliente pueda hacer uso de la aplicación.

Este proceso de asignación debe conllevar los siguientes pasos:

- Creación del registro del cliente en la tabla *tenant.*
- Creación del registro del usuario utilizando las credenciales ingresadas.
- Asignación de propiedad propietario al usuario que abrió la cuenta, lo cual permitirá que el usuario propietario sea capaz de modificar todas las configuraciones necesarias de la aplicación, para lograr una mayor personalización de la misma a los usuarios finales.
- Envío de correo notificando la URL de la aplicación del cliente al usuario propietario.
- Re-direccionamiento hacia la aplicación del cliente.

Una vez realizados los pasos anteriores, el usuario podrá ingresar sus credenciales en la pantalla de *Login* para hacer uso de la aplicación. El proceso que deberá realizar, se observa en la Figura 12:

Figura 12. **Asignación de subdominio**



Fuente: elaboración propia.

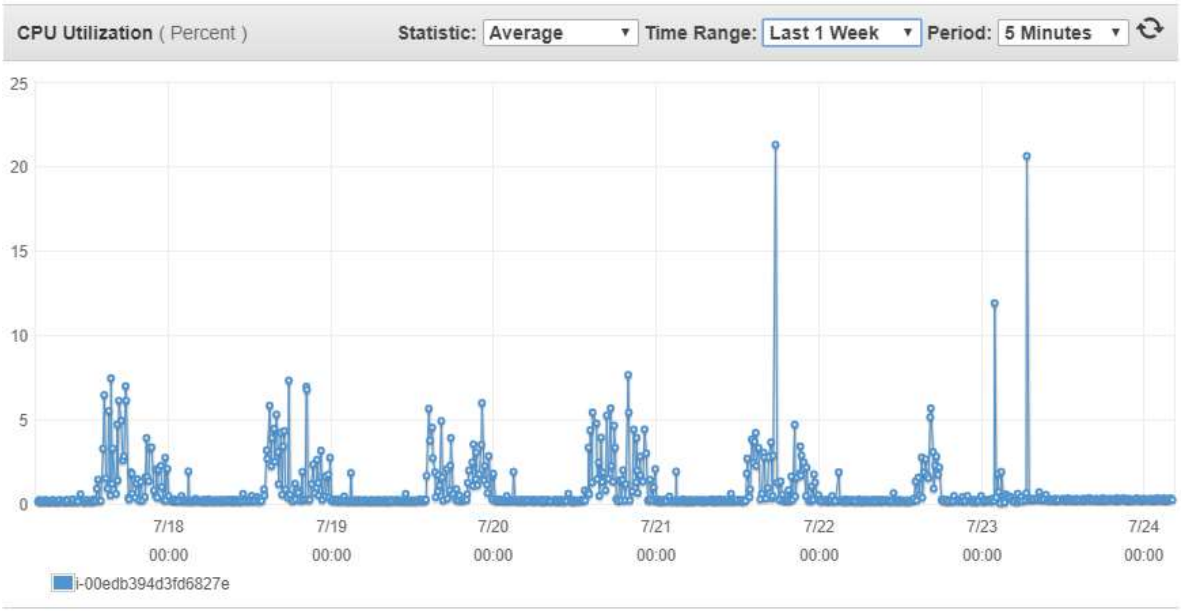
5.2.4. Evaluación de recursos computacionales

El correcto rendimiento de la aplicación es fundamental para proporcionar un buen servicio a los clientes y/o usuarios, quienes harán uso de la misma durante su jornada laboral. Por lo cual, el rendimiento de la aplicación debe escalar de forma automática para ajustarse a la demanda, de tal manera que si el número de usuarios aumenta o disminuye, la cantidad de servidores asignados a la aplicación debe de ajustarse, proporcionando así una aplicación con un rendimiento óptimo.

Para realizar el escalamiento automático, es importante la creación de controles que permitan monitorear el estado de los servidores, a partir de esto, decidir sobre la inclusión de un servidor adicional.

Utilizando tecnologías *Amazon Web Services*, la tarea de monitoreo se realiza, mediante la herramienta denominada *CloudWatch* que actúa en conjunto con *Auto Scaling* y ambas posibilitan el seguimiento de métricas de rendimiento como: uso de CPU, lecturas/escrituras en disco, bytes recibidos y enviados, entre otros. En la Figura 13, se observa una gráfica de ejemplo que despliega el porcentaje de utilización de CPU.

Figura 13. **Medición de utilización de CPU**



Fuente: Consola *CloudWatch* de *Amazon Web Services.*

En la anterior gráfica, se puede observar la tendencia de utilización de CPU de un servidor durante una semana, destacando de ella ciertos puntos que sobresalen de la tendencia normal. Dichos puntos superan el 20 % de uso de CPU para los días 21-07 y 23-07.

CloudWatch permite alertar o tomar acción al momento en que se cumpla una condición para una métrica de rendimiento, siendo para el presente estudio, una acción que corresponda al incremento de servidores como parte del conjunto de servidores que estarán detrás del balanceador de carga. La inclusión de un nuevo servidor, habilitará un servidor adicional al conjunto actual de servidores sobre el cual el balanceador distribuye las peticiones de usuario, logrando una mayor distribución, a través de todos los servidores, por ende, proporcionando una mejor experiencia de usuario.

En el presente estudio, se recomiendan los límites de la Tabla X, para las condiciones de métricas de rendimiento en caso de aumentar la demanda.

Tabla X. **Límites para incrementar el escalamiento horizontal**

Métrica	Límite	Durante	Acción
Promedio Utilización CPU	>= 80%	1 minutos	Agregar servidor
Promedio Utilización CPU	>= 70%	1 minutos	Agregar servidor
Promedio Utilización CPU	>= 60%	1 minutos	Agregar servidor
Promedio Utilización CPU	>= 50%	1 minutos	Agregar servidor

Fuente: elaboración propia.

En cuanto a la disminución de demanda, se recomiendan los límites de la Tabla XI para las condiciones de métricas de rendimiento.

Tabla XI. **Límites para reducir el escalamiento horizontal**

Métrica	Límite	Durante	Acción
Promedio Utilización CPU	<= 50%	1 minutos	Quitar servidor
Promedio Utilización CPU	<= 40%	1 minutos	Quitar servidor
Promedio Utilización CPU	<= 30%	1 minutos	Quitar servidor
Promedio Utilización CPU	<= 20%	1 minutos	Quitar servidor

Fuente: elaboración propia.

5.2.5. Definición de proceso de auto escalamiento

Acorde a la evaluación realizada por *CloudWatch* cada 1 minuto, se tomará una acción para satisfacer la demanda de los usuarios. En el caso de aumentar la cantidad de servidores del conjunto de servidores detrás del balanceador, es necesario que cada uno de los servidores que se agregue, tenga la configuración adecuada para que al momento de ser incluido en el conjunto, sea capaz de recibir peticiones del balanceador enviadas por los usuarios y responderlas satisfactoriamente.

5.2.5.2. Componentes de auto escalamiento

Para realizar este proceso de forma exitosa, se deben considerar diversos componentes que permitirán habilitar al nuevo servidor como parte del conjunto de servidores del balanceador. Los componentes a considerar son los siguientes:

- *Software* de terceros del servidor: es importante identificar que aplicaciones de terceros deben ser instaladas para el funcionamiento de la solución y su respectiva fuente de internet. Por ejemplo: Apache, NGINX, PHP, entre otros.
    - *Software*
    - Fuente
    - Versión
- Configuraciones de *software*: previo a la instalación de la solución en un servidor, se debe configurar el *software* de terceros para asegurar su correcto funcionamiento. Por ejemplo: archivos de configuración de Apache, PHP, base de datos, entre otros. Además de identificar las configuraciones, se debe definir el repositorio o fuente en donde se almacenarán para ser posteriormente descargadas. Algunas de las configuraciones por considerar son:
    - Parámetros de rendimiento, como uso de memoria, paginación, espacio en disco, etc.
    - Credenciales para la inicialización de servicios si aplica.
- Repositorio de aplicación *web*: se debe identificar algún sistema de control de versiones que permita la descarga de la solución para su respectiva instalación/configuración en el servidor. Para ello, es necesario identificar:
    - Fuente (URL).
    - Credenciales de acceso al repositorio.
- Configuraciones de solución: una vez descargada la aplicación en el nuevo servidor, será necesario que se apliquen las configuraciones de ambiente correspondientes de la solución, para asegurar que la aplicación se ejecuta

en un ambiente adecuado cumpliendo los lineamientos de seguridad. Para esto, se deben considerar configuraciones como:

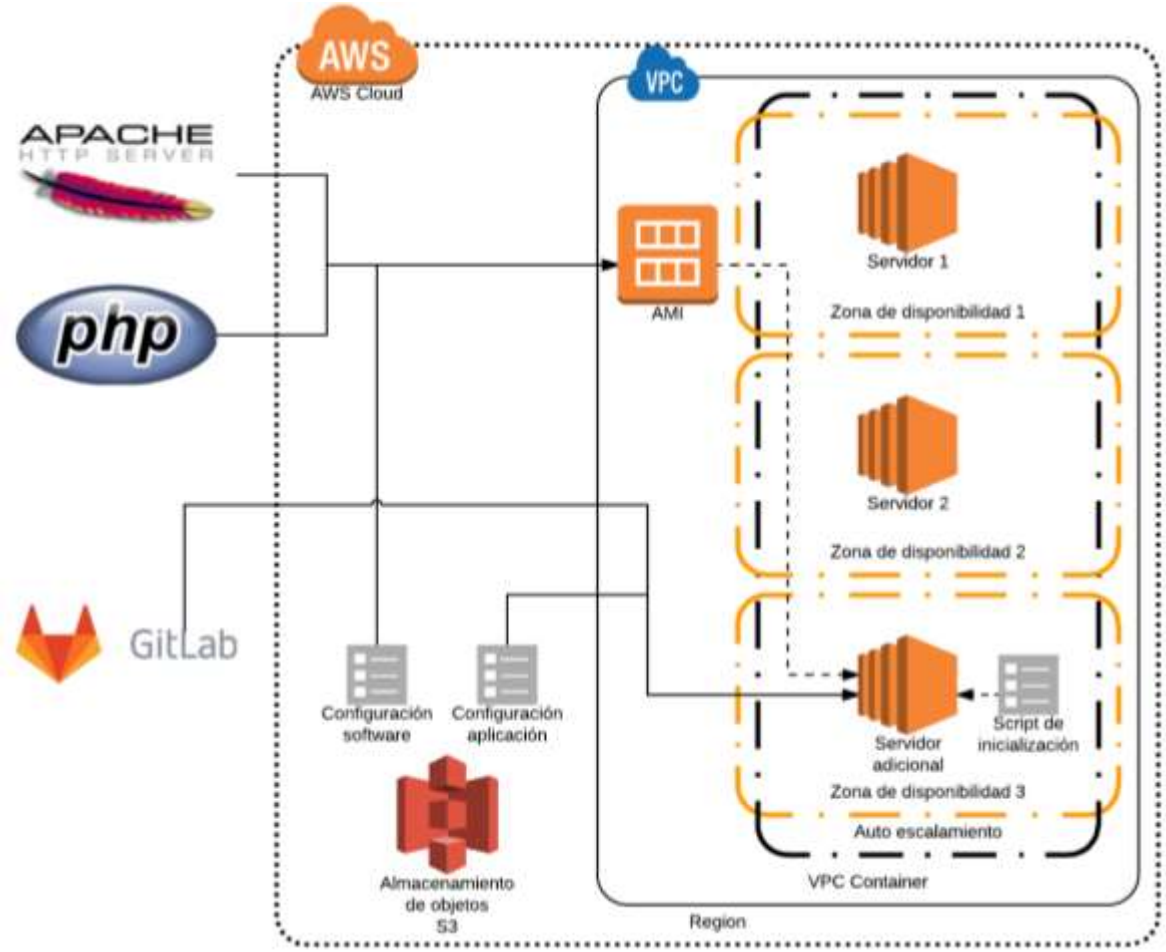
  - Credenciales de acceso a bases de datos.
  - Credenciales de acceso a almacenamiento de objetos.
  - Configuraciones de aplicación para un ambiente de producción.
  - Credenciales de acceso para envíos de correo, entre otros.

- Repositorio para archivos de configuración: para este repositorio se deberán identificar los siguientes elementos:
  - Archivos de configuración relacionados al paso 2.
  - Archivos de configuración relacionados al paso 3.
  - Fuente de archivos de configuración (repositorio).
  - Credenciales de acceso al repositorio.
- Script de inicialización: este será responsable de la descarga, instalación, configuración e inicialización de los anteriores elementos, y será ejecutado sobre el servidor que se desea agregar, permitiendo al servidor responder adecuadamente a las peticiones de los usuarios.
  - Forma de inyección/ejecución sobe el servidor a agregar.
- *AMI (Amazon Machine Image):* plantilla de máquina virtual a utilizar para los nuevos servidores que se crearán como parte del auto escalamiento. Esta plantilla deberá contener todo aquel *software* necesario para la correcta ejecución de la aplicación, tales son los componentes 1 y 2. El objetivo de esto, es permitir que la solución sea desplegada o instalada en un ambiente válido, correcto y seguro, lo cual se lograría a través de pruebas de vulnerabilidad del *software* a utilizar. Es decir, la plantilla de máquina virtual a utilizar para desplegar los nuevos servidores, deberá ser anteriormente validada para confirmar que no sufre de ninguna vulnerabilidad, y que cuenta con las configuraciones más óptimas para su correcto desempeño.

- Auto escalamiento: herramienta que brinda *Amazon Web Services* para ayudar a asegurar que se dispone del número correcto de servidores que puedan responder a la carga de la aplicación.

En la Figura 14, se despliega un diagrama de una posible y/o recomendada estructura de componentes, acorde a los mencionados anteriormente.

Figura 14. **Componentes de autoescalamiento**



Fuente: elaboración propia.

5.2.5.3. Creación de plantilla de máquina virtual (AMI)

Las tareas involucradas en este procedimiento, deben ser ejecutadas por un administrador de la consola de *Amazon Web Services,* ya que el administrador

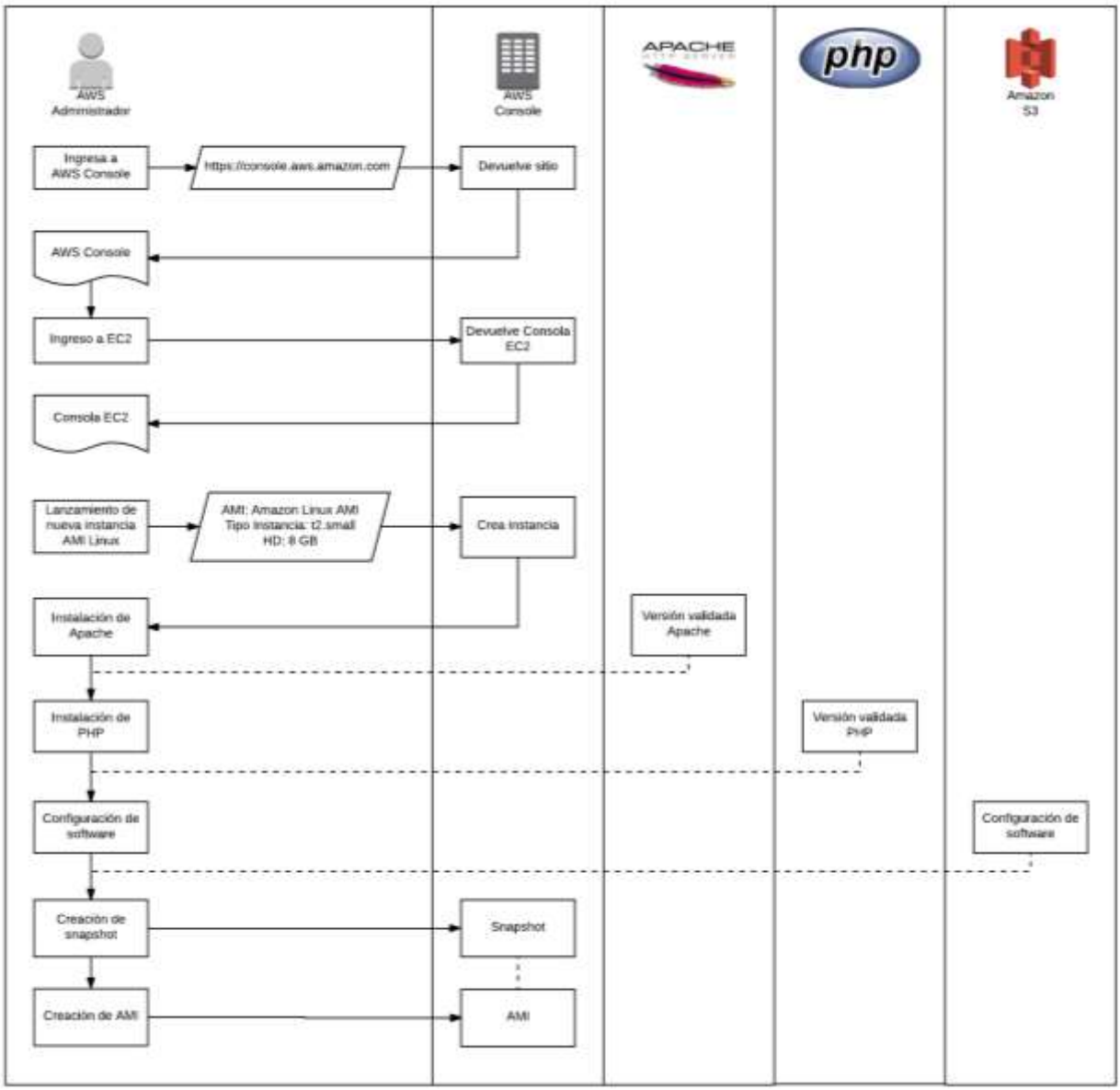
sería la persona responsable de la creación de nuevas instancias, debido a que es una acción que incurre en costos adicionales.

Luego de tener identificados los elementos de la sección anterior: *software*, fuente, versión y configuraciones, se podrán llevar a cabo las siguientes tareas que permitirán la creación de un AMI, que será la plantilla a utilizar para cada nuevo servidor a incluir en el auto escalamiento.

- Instalación de servidor *web,* a partir de repositorios válidos en internet.
- Instalación de librerías o componentes del lenguaje de servidor, a partir de repositorios válidos en internet.
- Configuración de servidor *web* y librerías del lenguaje de servidor.

En la Figura 15, se puede observar el flujo a realizar para la creación de la *AMI.*

Figura 15. **Creación de plantilla de máquina virtual (AMI)**



Fuente: elaboración propia.

5.2.5.4. Configuración grupo de auto escalamiento

Una vez se tenga definida el *AMI* a utilizar para el auto escalamiento, se debe crear el grupo de auto escalamiento, que básicamente consiste en definir

el proceso de escalamiento horizontal en el que se aumentará o disminuirá la cantidad de instancias utilizadas para soportar la demanda.

Para llevar a cabo este procedimiento, se deben realizar los siguientes pasos:

- Creación de configuración de lanzamiento.
    - Utilizar el *AMI* creada en la sección anterior.
    - Utilizar el tipo de instancia definido en la sección anterior.
    - Configurar el Script de inicialización como parte de *User Data,* el cual realizará:
        - Descarga de solución *web,* a partir del repositorio de control de versiones.
        - Configuración de solución *web.*
        - Inicialización de servicios.
    - Utilizar el almacenamiento definido en la sección anterior.
    - Utilizar el grupo de seguridad acorde a una subred pública. Esto se detallará en una sección posterior referente al diseño de red de servidores.
- Creación del grupo de auto escalamiento.
    - Definir las subredes en las cuales se deben crear las nuevas instancias. Es recomendable utilizar una subred por cada zona de disponibilidad, de tal manera que, se pueda tener brindar de alta disponibilidad de la aplicación, reduciendo las probabilidades de un fallo en el servicio.
    - Definir las políticas de escalamiento con base a los límites recomendados en la sección evaluación de recursos computacionales.
    - Definir las notificaciones a utilizar al momento de existir un cambio en el escalamiento: creación de nuevo servidor, finalización de servidor, falla durante la creación y/o falla durante la finalización de servidor.

- Creación de etiquetas para identificar más fácilmente a los servidores que son parte del auto escalamiento.

5.3. Cumplimiento de estándares de seguridad

5.3.1. Comunicación cifrada mediante HTTPS

Toda comunicación realizada en una aplicación *web,* que transfiera información confidencial, debe ser correctamente encriptada utilizando los protocolos de seguridad correspondientes. En este caso, dado que la comunicación se realiza vía protocolo HTTP (*Hypertext Transfer Protocol*), se recomienda utilizar el protocolo HTTPS (*Hypertext Transfer Protocol Secure*), el cual cifra la información utilizando el protocolo SSL (*Secure Socket Layer*) permitiendo una transferencia de datos segura, a través de internet.

Para hacer uso del protocolo HTTPS, se utiliza un certificado de seguridad SSL, los cuales pueden ser adquiridos con diversos proveedores como: www.interssl.com, www.gogetssl.com, www.instantssl.com, www.goddady.com, entre otros. Estos certificados son archivos que se configuran en el servidor *web*, y que se encargan de utilizar el protocolo SSL para cada una de las entradas y salidas que se realicen, a través del servidor *web*.

Para esta solución, se recomienda que el servidor *web* sea capaz de recibir peticiones en el protocolo HTTP; sin embargo, es ideal que el mismo servidor *web* se encargue de redireccionar las peticiones del protocolo HTTP al protocolo HTTPS, esto con el propósito de recibir todas las peticiones posibles por los usuarios, pero certificando una comunicación cifrada entre el cliente y el servidor.

Los certificados SSL utilizan dos llaves: una pública y otra privada. La llave pública es la que se transfiere al usuario final, y la llave privada es la que se almacena en el servidor. Esto permite que cada vez que el servidor envíe información al usuario final, la información se encripte utilizando la llave privada y el usuario final sea capaz de desencriptarla utilizando la llave pública. Un proceso similar se realiza cuando el usuario final envía información al servidor,

siendo en este caso, la llave pública la que permite encriptar la comunicación y la llave privada la que desencripta la comunicación en el servidor.

En el presente estudio, se propone una solución tecnológica que pueda ser utilizada por múltiples clientes/usuarios de forma concurrente, de tal manera que, la misma solución sea reutilizada para responder a muchas necesidades en paralelo. Esto requiere que la solución haga uso de subdominios para identificar a cada cliente que utiliza la aplicación, sin embargo, existen otros métodos para llevar esto a cabo, los cuales están fuera del alcance de este estudio.

Dado que la solución propuesta utiliza subdominios, el certificado de seguridad SSL, debe corresponder a un certificado comodín o *wildcard*, ya que estos permiten cifrar la comunicación en todos los subdominios existentes. Por ejemplo, en esta solución se propone utilizar la siguiente estructura de URL para cada cliente: *https://nombre.aplicacion.com*

En donde <nombre> corresponde a cada uno de los clientes que se tengan. Con el uso de este tipo de certificados se asegura que todos los subdominios utilizados usen el protocolo SSL, cifrando la comunicación entre usuario y servidor.

Adicional a la configuración aplicada con el certificado comodín SSL, se debe configurar el servicio de ruteo en *AWS Route 53,* para reenviar todas aquellas peticiones que se realicen sobre *.aplicacion.com hacia el balanceador de carga, ya que a nivel de aplicación es en donde se hace la distinción del subdominio y se filtra la información a desplegar.

En la Tabla XII, se muestran las configuraciones a realizar en AWS Route 53.

Tabla XII. **Configuraciones en AWS Route 53**

Configuraciones		
Nombre	**Tipo de registro**	**Valor**
aplicacion.com	A	DNS Balanceador de carga
*.aplicacion.com	A	ALIAS aplicacion.com

Fuente: elaboración propia.

5.3.2. Seguridad a nivel de aplicación

La seguridad es uno de los pilares más importantes que debe cubrir una solución *Software as a Service,* debido a que la información que se almacena corresponde a más de un cliente, y por lo tanto, debe ser correctamente resguardada. En esta sección, se cubren los aspectos más fundamentales que toda solución de *software* debe proveer como parte de la seguridad de la aplicación.

5.3.2.2. Autenticación

Es un proceso realizado al inicio del uso de una aplicación que permite confirmar las credenciales e identificar a la persona que estará utilizándola. De esta manera, se validará que la persona sea quien dice ser y que todas aquellas actividades realizadas dentro de la aplicación por ella, puedan ser rastreadas para futuras auditorías.

El proceso de autenticación requiere de los elementos descritos en la Tabla XIII:

Tabla XIII. **Elementos de autenticación**

Elemento	Descripción
Correo electrónico	Corresponde al correo electrónico del usuario que utilizará la aplicación.
Contraseña	Corresponde a un código de seguridad proporcionado inicialmente por el administrador de la aplicación, pero obligatoriamente modificado al primer inicio de sesión del usuario. La contraseña de todo usuario de la aplicación debe contar con altos estándares de seguridad. Se recomienda que para la política de contraseña se utilicen los siguientes lineamientos: • Estar compuesta de 8 o más caracteres • Tener al menos 1 letra mayúscula • Tener al menos 1 letra minúscula • Tener al menos 1 digito • Tener al menos 1 carácter no alfanumérico (, . ¡ ! ¿ ? *)

Fuente: elaboración propia.

Los elementos anteriormente descritos son básicos para todo proceso de autenticación; sin embargo, no proporcionan la suficiente seguridad para una aplicación que posea información confidencial de diversos clientes, ya que las credenciales pueden llegar a ser adivinadas por un tercero. Por tal razón, la aplicación debe ser capaz de proveer un método adicional de seguridad que se configure para el usuario que lo requiera.

En este caso, se recomienda el proceso de autenticación de dos factores, el cual consiste básicamente en configurar un paso adicional al método de autenticación descrito previamente. A dicho proceso se le suma una validación

realizada, a partir de la generación de un código mediante un dispositivo móvil, el cual debe ser introducido en la aplicación para su posterior validación. Con esta validación se estaría proporcionando un nivel de seguridad superior a la aplicación, ya que el proceso de autenticación consistiría de dos componentes:

Primer paso		**Segundo paso**
algo que el usuario sabe	**+**	**algo que el usuario tiene**
Credenciales		Código mediante móvil

Con la utilización de este mecanismo, se asegura que el usuario tenga conocimiento de cierto factor y que posea de otro factor para cumplir correctamente con el proceso de autenticación. Por lo tanto, se reducen considerablemente las probabilidades de permitir el acceso no deseado a la aplicación a un tercero.

El proceso completo de autenticación recomendado debe requerir los siguientes pasos:

- Creación de credenciales de usuario por parte del administrador, que incluye correo electrónico y contraseña.
- Ingreso del usuario a la aplicación utilizando las credenciales proporcionadas por el administrador.
- Solicitud de nuevas credenciales por parte de la aplicación al usuario cumpliendo la política de contraseña.
- Configuración de segundo factor de autenticación.
- Ingreso de usuario a la aplicación utilizando las credenciales configuradas y el segundo factor de autenticación.

5.3.2.4. Control de acceso basado en roles

Dado que la solución está enfocada en *freelancers* y se tienen diferentes planes de suscripción, como se observa en secciones posteriores, se debe

proveer de alta adaptabilidad a las necesidades de los mismos, quienes podrían requerir que diferentes usuarios puedan acceder a diferentes áreas de la aplicación, o bien, realizar diferentes acciones dentro de la aplicación.

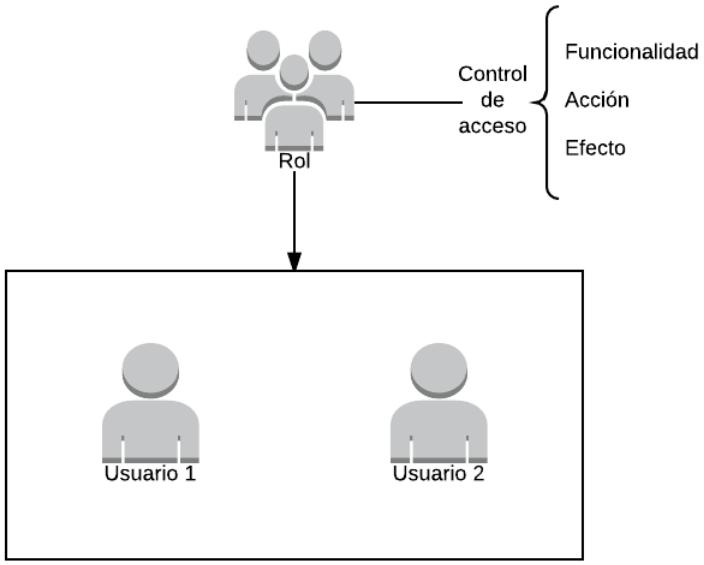
Los permisos dentro de una solución *Software as a Service* deben considerar varios elementos para proporcionar una mayor granularidad en el acceso a la información. Para lograr esto, se recomienda incluir al menos los siguientes elementos como parte de la configuración de control de acceso basado en roles:

- Funcionalidad: corresponde a la pantalla, reporte, función sobre la cual se desea dar permiso al sujeto. Por ejemplo, gestión de casos/proyectos, reporte de avance, entre otros.
- Acción: corresponde a la acción que se le permitirá al sujeto realizar sobre la funcionalidad. Por ejemplo, crear, editar o borrar.
- Efecto: indica si la acción asignada al sujeto sobre la funcionalidad será concedida o denegada.

Luego de tomar en consideración los anteriores elementos, se pueden realizar configuraciones tan granulares como sean posibles para cada uno de los roles creados dentro de la solución, permitiendo que cada uno de ellos cumpla con las políticas de acceso a la información acorde a los lineamientos del dueño de la solución.

Una vez creados y configurados cada uno de los roles dentro de la solución, estos se pueden asignar a los diferentes usuarios que existirán y harán uso de ella, obteniendo un control de acceso más granular y rápido de implementar sin requerir de un mantenimiento complejo.

Figura 16. **Control de acceso basado en roles**



Fuente: elaboración propia.

5.3.2.3. Variables de sesión

En toda aplicación de *software* se manejan variables de sesión que almacenan información importante para el correcto funcionamiento y personalización de la aplicación, acorde al usuario que esté haciendo uso de ella. Algunos de los elementos que se llegan a guardar como parte de las variables de sesión son:

- Usuario: nombre, apellido, correo electrónico, cuenta.
- Cuenta (*tenant*): nombre.
- Rol: todos los permisos asignados al rol.
    - Funcionalidad
    - Acción
    - Efecto

Idealmente las variables de sesión deben ser correctamente almacenadas en el servidor para evitar acceso a las mismas y que pueda darse fuga de información. Debido a esto, se recomienda aislar completamente el manejo de

las variables de sesión del servidor, y utilizar un gestor adicional que las almacene.

Siguiendo con la premisa que la solución se encontrará en la nube y que se pueden utilizar servicios administrados por *Amazon Web Services*, se propone el uso de *Amazon ElastiCache* que es un servicio de almacenamiento en memoria o en cache, que aumenta el rendimiento de aplicaciones *web,* debido a que se evita el almacenamiento en disco.

Amazon ElastiCache proporciona dos motores de memoria:

- Memcache
- Redis

Para este caso, se recomienda el uso de Redis, debido a la alta disponibilidad que se puede lograr con dicho motor; además de su rendimiento y confiabilidad proporcionada.

Como parte de la configuración a realizar en *Amazon ElastiCache,* se deben considerar las propiedades de la Tabla XIV, ya que tienen la misma importancia que la configuración de *Amazon RDS* mencionada en secciones anteriores.

Tabla XIV. **Propiedades de Amazon ElastiCache**

Propiedad	Valor
Propiedades Redis	
Número de réplicas	1
Propiedades avanzadas Redis	
Multi-AZ despliegue	Verdadero
VPC	VPC creada en sección posterior (10.0.0.0/16)
Subredes	Subred privada – base de datos (primaria) Subred privada – base de datos (réplica)
Zona de disponibilidad	Primaria: Subred privada – base de datos (primaria) Réplica 1: Subred privada – base de datos (réplica)
***Security groups* (Firewall)**	Firewall privado – base de datos
Respaldo	
Habilitar respaldos automáticos	Verdadero
Período de retención de respaldo	1 día
Hora de inicio	00:00 UTC
Duración	1 hora

Fuente: elaboración propia.

5.3.2.4. Web application firewall

Finalmente, se debe proteger la aplicación en cada una de las peticiones que sean realizadas sobre la misma, debido a que existen múltiples técnicas que pueden ser utilizadas para inyectar código malicioso y vulnerar la seguridad de la información.

Algunas de las técnicas más utilizadas para quebrantar la seguridad de una aplicación son:

- *Cross-site scripting*
- Inyección SQL

Ambas técnicas se pueden presentar en las peticiones que los usuarios realizan sobre la aplicación, causando fallas y en algunos casos se logra el robo de información valiosa. Dado esto, es importante que se realice la configuración de un *Web application firewall,* el cual se debe situar al frente del balanceador de carga de la aplicación a nivel de *backend.*

Amazon Web Services proporciona un servicio capaz de detectar aquel código malicioso que forma parte de las peticiones de una aplicación *web*. Este servicio se llama *AWS WAF* (*Web Application Firewall*), el cual se configura con una serie de condiciones capaces de monitorear cada petición, a partir de diversas validaciones y determinar si la petición contiene código malicioso. Una vez validada la petición, *AWS WAF* se encarga de permitir o bloquear la petición con base a las condiciones y reglas definidas en su configuración. *AWS WAF* se compone de tres elementos:

- Condiciones: definen características básicas que se desean monitorear.
- Reglas: con base a las condiciones previas, se construyen reglas con mayor complejidad para determinar que ayudarán a determinar si una petición se permite, se bloquea o se cuenta.
- *Web ACL (Access Control List):* utilizando las reglas previas, se determina la acción a realizar: permitir, bloquear o contar.

En la Tabla XV, se puede observar la configuración a realizar en el servicio *AWS WAF. S*e recomienda de esta manera, debido a que si una condición contiene más de un filtro, cualquiera de ellos puede cumplirse para determinar la

acción a realizar, a diferencia de si se crearán diferentes condiciones, se tendrían que cumplir todas ellas para determinar la acción.

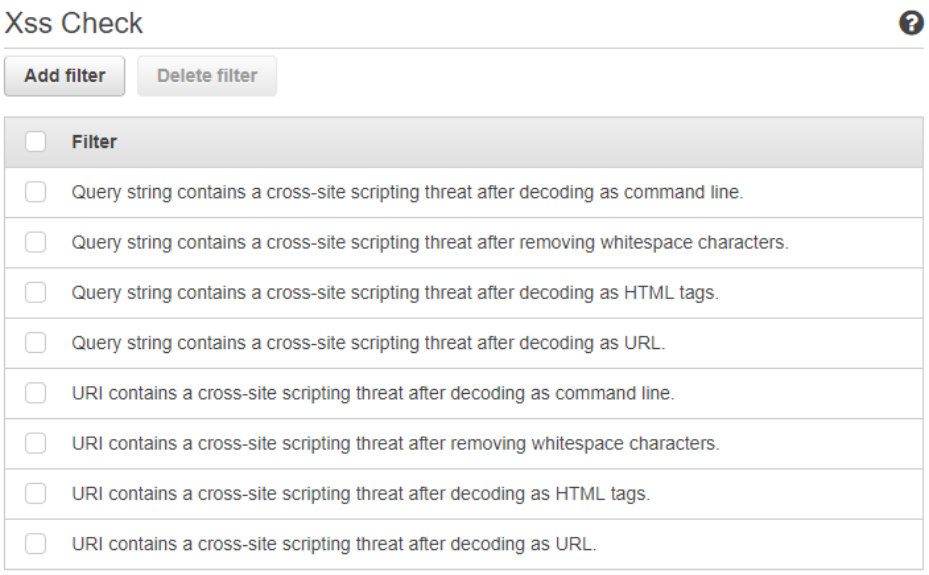
Tabla XV. **Configuración de AWS WAF**

***Web* ACL**	**Regla**	**Condición**	**Filtro**	
			Parte de la petición	**Transformación**
WAF	Xss Rule	Xss Check	Query string	Command line
			Query string	Whitespace characters
			Query string	HTML tags
			Query string	URL
			URI	Command line
			URI	Whitespace characters
			URI	HTML tags
			URI	URL
	SQLi Rule	SQLi Check	Query string	Command line
			Query string	Whitespace characters
			Query string	HTML tags
			Query string	URL
			URI	Command line
			URI	Whitespace characters
			URI	HTML tags
			URI	URL

Fuente: elaboración propia.

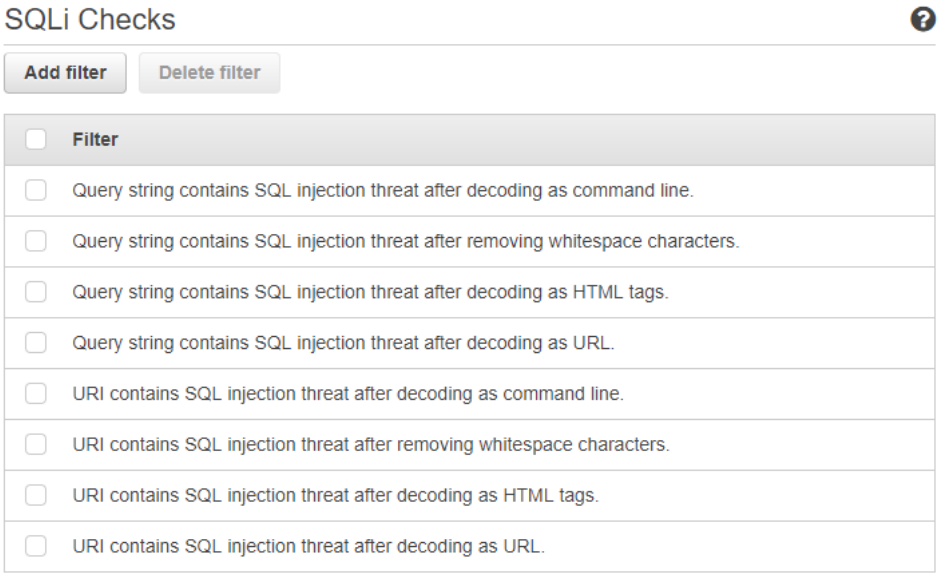
En la Figura 17, se pueden observar los filtros utilizados para la condición de *Cross-site scripting,* y en la Figura 18, se observan los filtros para la condición de Inyección SQL.

Figura 17. **Filtros para *cross-site scripting***



Fuente: elaboración propia.

Figura 18. **Filtros para inyección SQL**

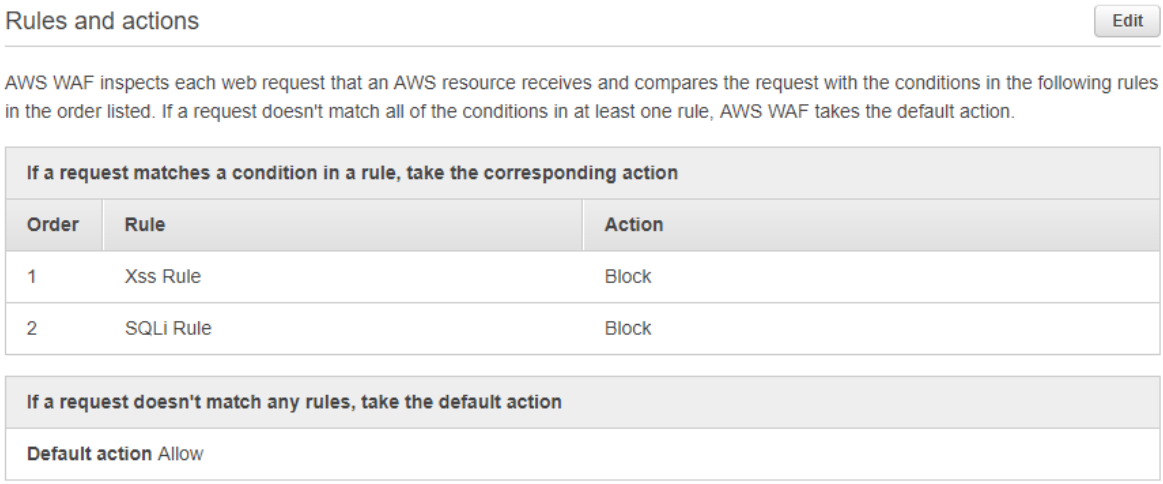


Fuente: elaboración propia.

Una vez se tengan configuradas las condiciones y reglas a utilizar, se debe configurar el *Web ACL,* en el cual se le indica la acción a realizar. Tomando en

consideración que todos los filtros son para detectar la existencia de código malicioso, la acción a realizar por Web ACL deberá corresponder al bloqueo de la petición. Esto se puede observar en la Figura 19.

Figura 19. **Configuración de *Web Application Firewall***



Fuente: elaboración propia.

Al *Web* ACL se le deberá asociar el Application Load Balancer que se sitúa al frente de los servidores de aplicación en la subred privada – *backend*, para que toda petición que llegue a los servidores de aplicación, sea cuidadosamente monitoreada y validada previo a su entrega y evitar cualquier daño que pueda ser ocasionado sobre la aplicación o la base de datos.

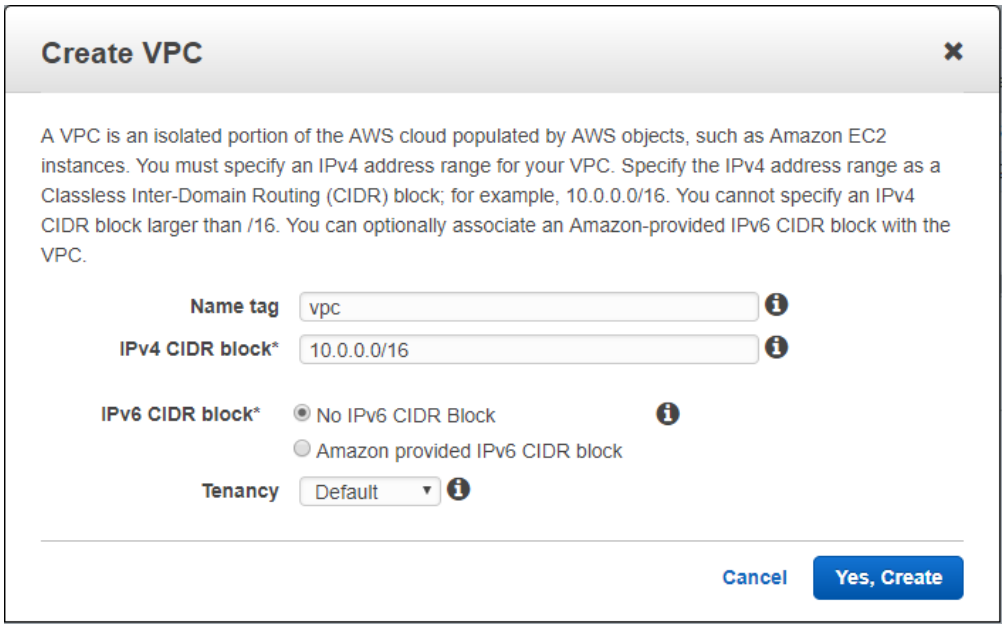
5.3.3. Diseño de red de servidores

La infraestructura en la que resida la solución debe tener los mejores controles de seguridad, con el propósito de disminuir los riesgos de acceso no deseado a información confidencial. Para esto, la red en la que se encuentren los servidores de la solución, debe estar correctamente configurada, y las subredes que posea deben estar aisladas entre sí, para que no se puedan lograr saltos entre los servidores que quiebren la seguridad de la solución.

En el caso de *Amazon Web Services,* la red en donde residen los componentes de una solución se llama *Virtual Private Cloud,* que corresponde a una red virtual alojada en la nube de *Amazon Web Services* y se encuentra aislada de forma lógica del resto de la nube, simulando una red privada.

Para el presente estudio, se sugiere la configuración de la Figura 20 para la *VPC*.

Figura 20. **Creación de VPC**



Fuente: elaboración propia.

5.3.3.1. Configuración de grupo de subred

Para un correcto funcionamiento de la infraestructura de la solución, es preciso que se cree un grupo de subred que consistirá de las diferentes subredes disponibles en la aplicación en las cuales se colocarán las bases de datos.

Este grupo de subred se utilizará para la configuración de replicación de las bases de datos, que se tratará en secciones posteriores. La configuración de dicho grupo de subred se puede apreciar en la Tabla XVI.

Tabla XVI. **Grupo de subred**

Tipo de red	Capa de aplicación	Rango de IP	Zona de disponibilidad
Privada	Base de datos (primaria)	10.0.3.0/24	Zona de disponibilidad 2
Privada	Base de datos (secundaria)	10.0.6.0/24	Zona de disponibilidad 1
Privada	Base de datos (uso futuro)	10.0.9.0/24	Zona de disponibilidad 3

Fuente: elaboración propia.

5.3.3.2. Configuración de subredes

Dado que la solución debe garantizar seguridad en todos los niveles, es primordial que el diseño de la red considere el aislamiento de los servidores acorde a su función. Con base a este aislamiento, los servidores residirían en diferentes subredes dependiendo del rol que ejecuten dentro de la solución. Residiendo, por ejemplo, los servidores *web* en una subred pública que posibilite su acceso, a través de internet a la solución; por otro lado, los servidores de aplicación residirían en una subred privada, a la cual únicamente tendrían acceso los servidores públicos en puertos específicos.

De esta manera, se evitaría el acceso desde internet a los servidores de aplicación e incluso a las bases de datos, asegurando completamente la integridad de los mismos.

La cantidad de subredes a crear puede corresponder directamente a la cantidad de capas que tenga la arquitectura de la aplicación, con el objetivo de

aislar cada una de ellas para mayor seguridad. En este documento se consideran tres capas:

- *Frontend*: aplicación *web* que hará entrega de la solución. Contendrá todos los elementos html, javascript, css.
- *Backend*: contendrá todos aquellos servicios de aplicación que permitan entregar la información al *front end.*
- Base de datos: corresponde a las bases de datos en donde se almacenará la información de los usuarios.

Con base a lo anterior, se propone la creación de tres subredes como se puede observar en la Tabla XVII.

Tabla XVII. **Subredes de solución**

Tipo de red	Capa de aplicación	Rango de IP	Acceso desde
Pública	*Frontend*	10.0.1.0/24	Internet
Pública	*Frontend*	10.0.4.0/24	Internet
Pública	*Frontend*	10.0.7.0/24	Internet
Privada	*Backend*	10.0.2.0/24	Subred pública
Privada	*Backend*	10.0.5.0/24	Subred pública
Privada	*Backend*	10.0.8.0/24	Subred pública
Privada	Base de datos (primaria)	10.0.3.0/24	Subred privada – *Backend*
Privada	Base de datos (secundaria)	10.0.6.0/24	Subred privada – *Backend*
Privada	Base de datos (uso futuro)	10.0.9.0/24	Subred privada – *Backend*

Fuente: elaboración propia.

5.3.3.3. Configuración de firewall

Toda solución de *software* requiere de la configuración de un *firewall* que asegure una zona crítica en la que se encuentran elementos de alta importancia como: servidores, bases de datos, almacenamiento de objetos, entre otros.

En este estudio, se considera tantos *firewalls* como subredes existan en la solución, por lo tanto, uno de ellos se encontrará al frente de la aplicación y filtrará todas las peticiones realizadas, permitiendo únicamente el acceso a un servidor *bastion,* que fungirá como intermediario entre el usuario y la red de servidores internos. El segundo *firewall* se encontrará protegiendo a aquellos servidores que se localicen dentro de la subred privada de *Backend*. Y por último, un tercer *firewall* que se encargará de proteger a la base de datos en donde se almacene la información de los usuarios.

De esta manera, se imposibilita el acceso directo a los servidores de aplicación o a las bases de datos desde el internet. Por lo tanto, si se desea acceder a algún servidor de la aplicación, primero se debe acceder a un servidor *bastion* y posteriormente se accederá al servidor deseado.

El análisis o detalle del servidor *bastion*, queda fuera del alcance de este estudio, debido a que forma parte de una configuración genérica para todas aquellas redes que desean un mayor nivel de seguridad.

La configuración a realizar para cada uno de los *firewalls* debe ser realizada en *Amazon Web Services,* mediante la creación de un *Security group,* que posteriormente debe ser asignado a cada uno de los servidores que se alojen en la subred correspondiente.

***Firewall* público**

Considerando que será una solución *web* accedida mediante un navegador, se pueden considerar las siguientes reglas de la Tabla XVIII como parte del

firewall a configurar para los servidores públicos, que serán todos aquellos que se encuentren en las subredes públicas.

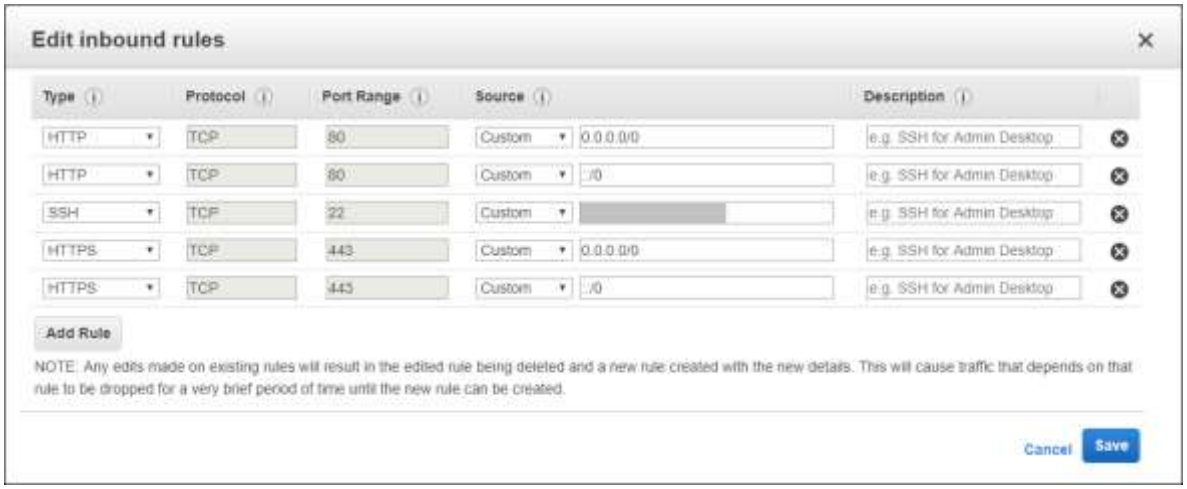
Tabla XVIII. **Configuración de *firewall* público**

Tipo de regla	Protocolo	Puerto	Origen
HTTP	TCP	80	Todas
HTTPS	TCP	443	Todas
SSH	TCP	22	IP específica de la red local interna

Fuente: elaboración propia.

El *Security group* debe quedar configurado según la Figura 21.

Figura 21. **Configuración de *firewall* público**



Fuente: elaboración propia.

Firewall* privado – *Backend

Los servidores ubicados en las subredes privadas de *backend* deben ser únicamente accedidos por aquellos servidores que se encuentren en las subredes públicas, por lo cual, la configuración del *firewall* difiere respecto al de la sección anterior.

En este caso, se debe especificar que el origen de cada una de las reglas a definir, corresponde a las direcciones de las subredes públicas. Además de considerar que los puertos a aplicaciones a las que se desea acceder a estos servidores son:

- Apache o cualquier otro servidor *web.*
- Acceso a servidores: SSH o RDP, dependiendo de la plataforma. En el presente estudio, se propone Linux, por lo tanto, se usará SSH y el respectivo puerto 22.

En la Tabla XIX, se puede observar la configuración del *firewall* privado correspondiente al *backend*.

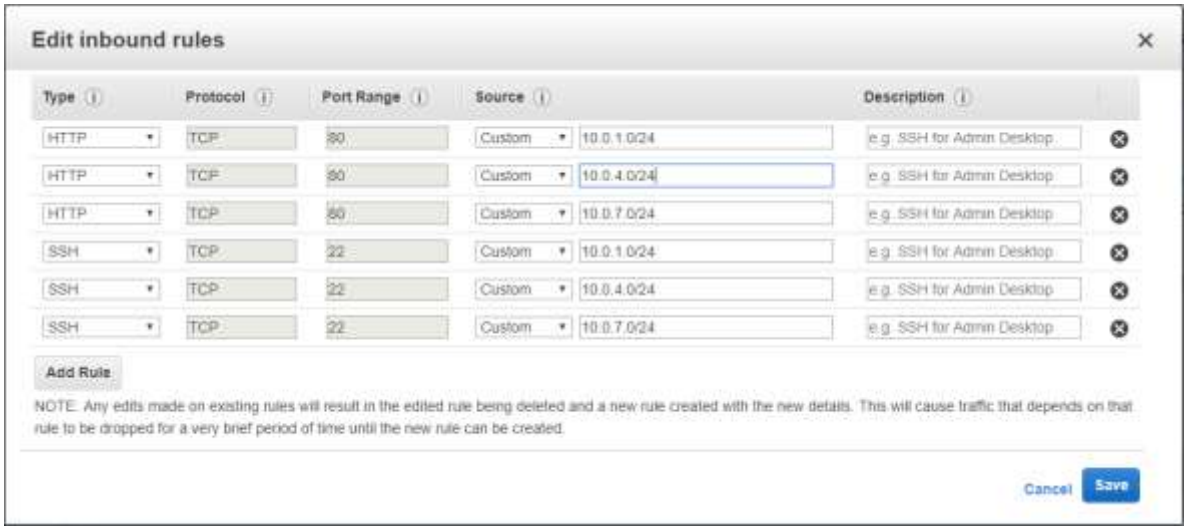
Tabla XIX. **Configuración de *firewall* privado *backend***

Tipo de regla	Protocolo	Puerto	Origen
HTTP	TCP	80	IP's subredes públicas
HTTPS	TCP	443	IP's subredes públicas
SSH	TCP	22	IP's subredes públicas

Fuente: elaboración propia.

La configuración del *Security group* correspondiente, se puede observar en la Figura 22.

Figura 22. **Configuración de *firewall* privado *backend***



Fuente: elaboración propia.

***Firewall* privado – Base de datos**

En el presente estudio, se utilizará como base de datos un servicio de *Amazon Web Services* llamado *Relational Database Service (RDS),* el cual corresponde a una base de datos completamente administrada por *Amazon,* permitiendo su rápido escalamiento, mantenimiento, alta disponibilidad, entre otras funcionalidades.

La base de datos propuesta para este estudio es MySQL y será accedida desde los servidores de aplicación, que se encuentran ubicados en la subred privada de *backend*.

El origen de cada una de las reglas a definir para este *firewall*, corresponde a la dirección de la subred privada de *backend*, con el puerto a la base de datos MySQL desea acceder. El puerto a habilitar corresponde a:

- Base de datos: MySQL, con su respectivo puerto 3306.

En la Tabla XX, se puede observar la configuración del *firewall* privado correspondiente al *backend*.

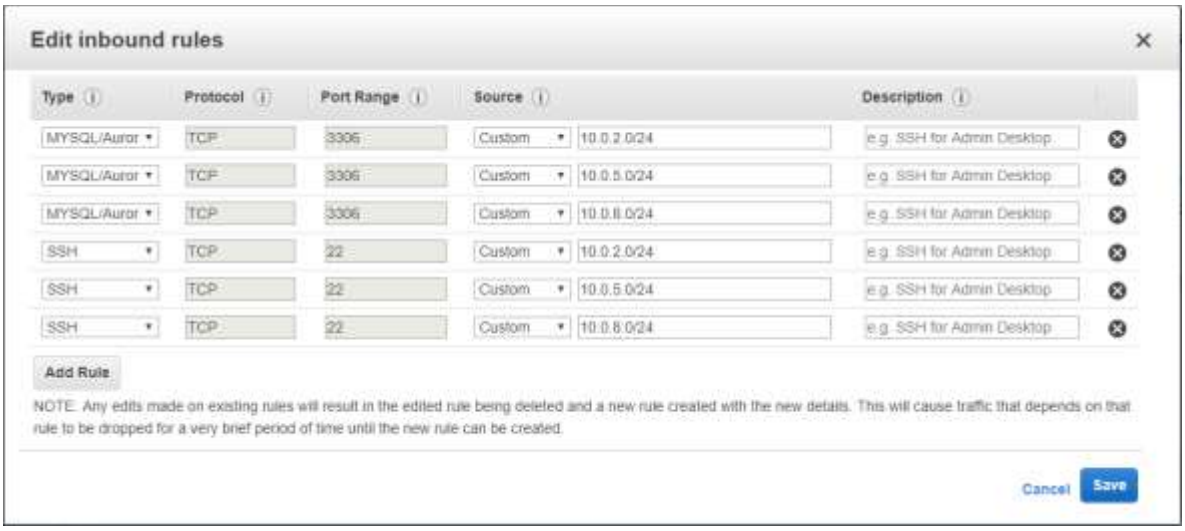
Tabla XX. **Configuración de *firewall* privado base de datos**

Tipo de regla	Protocolo	Puerto	Origen
MySQL	TCP	3306	IP's subredes privadas *backend*
SSH	TCP	22	IP's subredes privadas *backend* *Opcional si se tiene servidores en la subred

Fuente: elaboración propia.

La configuración del *Security group* correspondiente, se puede observar en la Figura 23.

Figura 23. **Configuración de *firewall* privado base de datos**



Fuente: elaboración propia.

5.3.3.4. Configuración de tablas de ruteo

Las tablas de ruteo son importantes para configurar correctamente la comunicación entre las subredes de la solución. Como se definió en la sección anterior, se propone una subred para cada una de las capas de la aplicación, dando como resultado, tres subredes en total.

Por esta razón, cada una de las subredes debe poseer su propia tabla de ruteo y así permitir o denegar el acceso a los elementos que se alojen en ella.

Tabla de ruteo – Pública

La subred pública debe tener asignada una tabla de ruteo que apunte hacia el *Internet Gateway,* con el propósito de tener acceso a internet para que todos los usuarios puedan utilizar la aplicación.

Para esto, es necesario que se cree una tabla de ruteo con acceso público, y se utilice una configuración como se observa en la Figura 24.

Figura 24. **Tabla de ruteo de subred pública**

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	igw-b9daa0d0	Active	No

Fuente: elaboración propia.

Tabla de ruteo – Privada

La subred privada en donde se alojen los servidores de aplicación, debe tener asociada una tabla de ruteo que permita únicamente la comunicación entre las mismas subredes de la red y el *NAT Gateway* definido en sección previa. Tomando como base que ambas reglas, se debe considerar la tabla de ruteo que se muestra en la Figura 25.

Figura 25. **Tabla de ruteo de subred privada *backend***

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	nat-04d2c3ef0ba5d6efc	Active	No

Fuente: elaboración propia.

5.3.4. Seguridad a nivel de base de datos

Acorde a la arquitectura definida en secciones previas, la solución utilizará el servicio *Relational Database Service (RDS),* que hace entrega de un gestor de base de datos completamente administrado por *Amazon Web Services*, reduciendo la complejidad de administración de base de datos.

El servicio *Amazon RDS* provee de diversas facilidades en cuanto a la gestión de base de datos, entre ellas:

- Fácil administración
- Escalabilidad
- Disponibilidad
- Durabilidad
- Rapidez y
- Seguridad

Las anteriores características de *Amazon RDS* hacen que varias de las tareas de mantenimiento de una base de datos sean ejecutadas de forma automatizada, sin intervención humana, por *Amazon Web Services.*

En las secciones posteriores, se detallan algunas de las tareas relevantes de una base de datos concernientes a la seguridad que se les debe aplicar a las mismas. Adicionalmente a dichas secciones, es importante mencionar que la base de datos debe ser alojada en la subred privada de base de datos.

La base de datos a configurar debe tener en sus propiedades de red y seguridad las opciones detalladas en la Tabla XXI.

Tabla XXI. **Propiedades de red y seguridad de base de datos**

Propiedad	Valor
VPC	VPC creada en sección previa (10.0.0.0/16)
Accesible públicamente	No

Zona de disponibilidad	Subred privada – base de datos (primaria)
***Security group* (Firewall)**	Firewall privado – base de datos

Fuente: elaboración propia.

5.3.4.1. Encriptación de base de datos

La información es uno de los principales activos de toda empresa, debido a la alta importancia que representa para su constante evolución y adaptación a las nuevas necesidades del mercado.

Utilizando el servicio *Amazon RDS,* fácilmente se puede encriptar la información almacenada en la base de datos, consiguiendo un nivel superior de seguridad dado el cifrado aplicado, el cual impide la lectura de los datos.

Para aplicar encriptación de base de datos, basta con realizar la configuración de la Tabla XXII al momento en que se está creando el recurso en *Amazon RDS*.

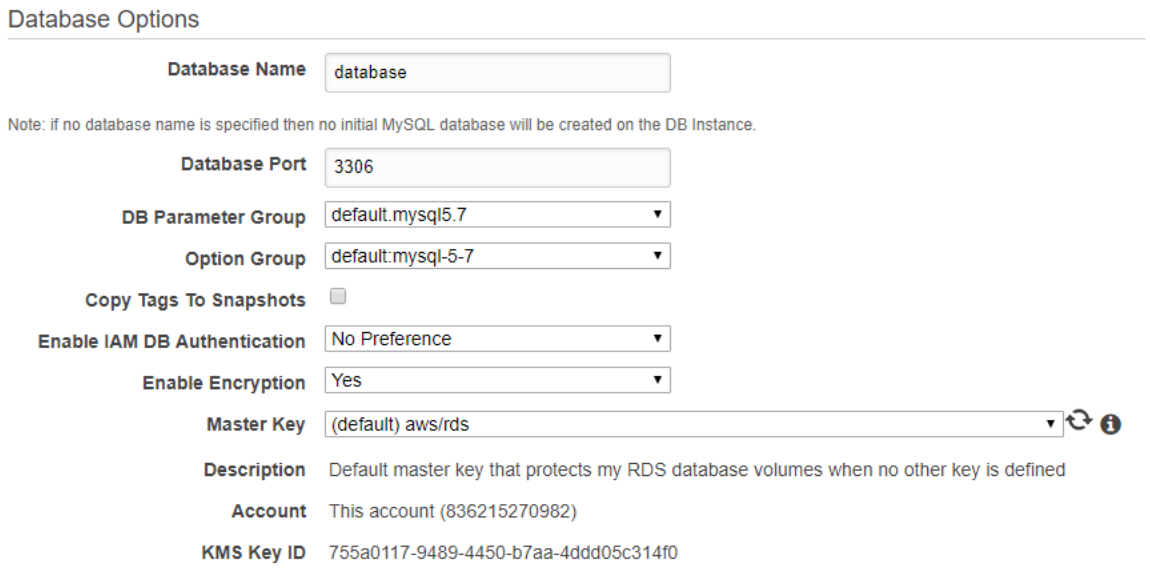
Tabla XXII. **Propiedades de encriptación**

Propiedad	Valor
Habilitar encriptación	Sí
Clave maestra	(default) aws/rds

Fuente: elaboración propia.

Las propiedades de encriptación se encuentran en las opciones de base de datos mientras se crea la misma. En la Figura 26, se despliegan las opciones que se deben configurar correspondientes a la Tabla XXII, siendo la clave maestra la utilizada por defecto por *Amazon Web Services*. Posterior a la configuración, todos los datos que se almacenen en la base, serán encriptados utilizando la clave maestra y no podrán ser desencriptados por terceros.

Figura 26. **Propiedades de base de datos**



Fuente: elaboración propia.

5.3.4.2. Replicación de base de datos

La seguridad de la información también está en función de la alta disponibilidad de la misma, ya que mientras se cuente con una copia en tiempo real, el usuario podrá tener acceso a ella sin ninguna interrupción.

Con *Amazon RDS* también se puede lograr una replicación síncrona, utilizando la funcionalidad de despliegue *Multi-AZ*, que corresponde a un despliegue en múltiples zonas de disponibilidad. Cuando una base de datos *Amazon RDS* tiene este tipo de configuración, se crea una replicación síncrona en una zona de disponibilidad diferente.

Por lo tanto, se obtienen dos bases de datos:

- Primaria: en zona de disponibilidad correspondiente a la subred privada – base de datos.
- Secundaria: en zona de disponibilidad correspondiente a la segunda subred privada – base de datos. Esta base de datos pasará a fungir como

base de datos primaria, una vez la base de datos primaria presente fallas o no esté disponible. Este procedimiento será realizado de forma automática por *Amazon RDS,* sin necesidad de realizar algún procedimiento manual por parte de los administradores de *Amazon Web Services*.

La única configuración que se debe realizar para habilitar el despliegue *Multi-AZ,* se encuentra en las especificaciones de la instancia, según Tabla XXIII.

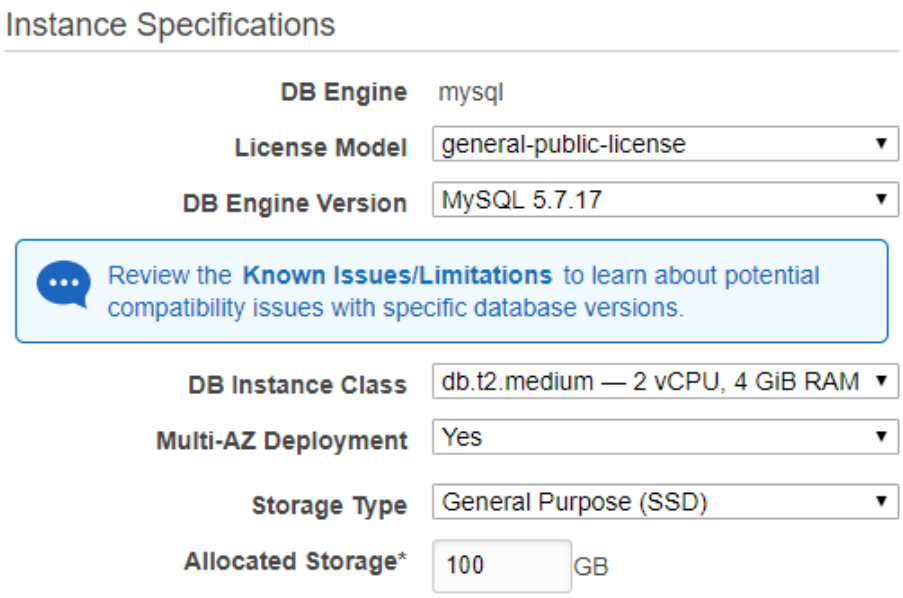
Tabla XXIII. **Especificaciones de instancia**

Propiedad	Valor
Despliegue Multi-AZ	Sí

Fuente: elaboración propia.

La configuración del despliegue *Multi-AZ* se realiza al momento en que se crea la base de datos *Amazon RDS,* quedando de la siguiente manera.

Figura 27. **Especificaciones de instancia**



Fuente: elaboración propia.

5.3.4.3. Procedimientos de respaldo de base de datos

Amazon RDS también cuenta con una funcionalidad sumamente efectiva para la creación automática de respaldos que realiza copias de seguridad diariamente a determinada hora, según la configuración definida para la instancia de base de datos.

Dado que en el presente estudio, se sugiere la creación de una instancia de base de datos utilizando despliegue *Multi-AZ,* la creación automática de respaldos no tiene ninguna afección sobre el rendimiento de la instancia, ya que las operaciones I/O sobre ella, no se suspenden en ningún momento; a diferencia de una instancia con un despliegue *Single-AZ,* el cual únicamente tiene una instancia primaria sin contar con una instancia de replicación.

Adicionalmente a los respaldos realizados diariamente sobre la instancia, *Amazon RDS* captura *transaction logs,* los cuales se encargan de registrar cada una de las actualizaciones realizadas sobre la instancia. De esta manera, se puede lograr una recuperación en cualquier punto del tiempo, debido a que *Amazon RDS* combina tanto el respaldo automático diario con los *transaction logs* para realizar una recuperación total de la instancia.

Por defecto, *Amazon RDS* habilita un período de retención de respaldos de siete días, almacenando el respaldo diario de los últimos siete días para su posible recuperación, después de este período, cada respaldo es eliminado automáticamente. Las propiedades de configuración de la instancia, se detallan en la Tabla XXIV.

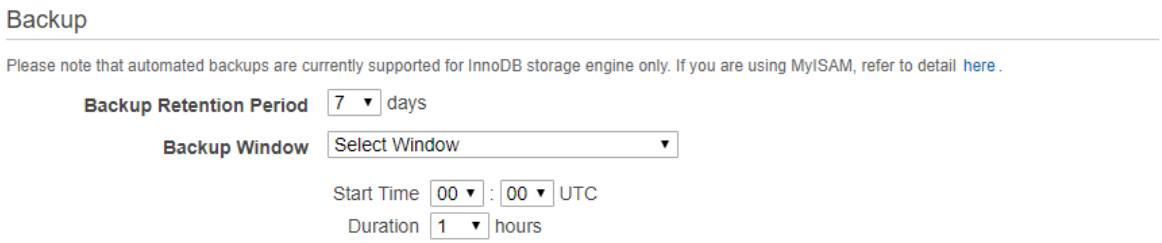
Tabla XXIV. **Propiedades de respaldo**

Propiedad	Valor
Período de retención de respaldo	7 días
Hora de inicio	00:00 UTC
Duración	1 hora

Fuente: elaboración propia.

La configuración resultante sería acorde a la Figura 28.

Figura 28. **Propiedades de respaldo**



Fuente: elaboración propia.

5.3.5. Política de privacidad

Como parte fundamental de una solución *Software as a Service,* se debe considerar la política de privacidad, que se utilizará como parte del servicio brindado, la cual debe transmitir todos los lineamientos que serán utilizados sobre la información confidencial que sea ingresada por los clientes en la aplicación, permitiendo ofrecer un servicio seguro y confiable como parte de los estándares de seguridad.

En la sección de Anexos, se presenta una política de seguridad que cubre los elementos más importantes de una solución *Software as a Service*.

5.4. Modelo de negocio

5.4.1. Identificación del modelo de negocio

Existen dos grandes modelos de negocio para las soluciones de *software*, uno de ellos es el conocido *On-premise* que consiste en el licenciamiento del *software* que se desea utilizar más una comisión de uso que se debe pagar periódicamente. Por otro lado, el modelo de negocio que hoy en día resulta ser más rentable para los clientes, que es el modelo *Software as a Service,* el cual consiste en conceder el derecho de uso de una solución a los clientes, una vez se haya realizado el pago correspondiente hacia el proveedor.

Este modelo de negocio resulta bastante atractivo para ambas partes: el cliente y el proveedor, ya que el cliente se beneficia con todas las ventajas que se pueden observar en la Tabla XXV.

Tabla XXV. **Comparativo on-premise y SaaS**

Incluye	On-premise	SaaS
Infraestructura	X	✓
Sistema operativo	X	✓
Instalación de *software*	X	✓
Seguridad integrada	X	✓
Disponibilidad	X	✓
Respaldo	X	✓
Mantenimiento	X	✓
Escalabilidad	X	✓
Rápida implementación	X	✓

Bajo costo	X	✓
Mejoras constantes	X	✓

Fuente: elaboración propia.

Tal como se puede observar en la comparativa de soluciones *On-premise* y *Software as a Service,* el cliente incurre en menos gastos cuando hace uso de una solución *SaaS,* ya que no requiere de invertir capital en la compra de servidores, *software*, mantenimiento, implementación, entre otros. Adicionalmente, el cliente tiene mayor flexibilidad en cuanto a la evaluación de una solución que cubra de mejor manera sus necesidades, debido a que con una solución *SaaS,* el cliente no adquiere ningún tipo de compromiso a largo plazo con el proveedor, por el contrario, tiene la flexibilidad de terminar el uso de cierta solución y evaluar otra en el momento que desee.

Estas ventajas le brindan al cliente muchos beneficios, tanto económicos como operativos, permitiéndole al cliente enfocarse en el día a día de su negocio, y delegar todas aquellas tareas administrativas del *software* al proveedor, haciendo más efectiva y productiva la colaboración de su equipo técnico en proyectos clave para agregar valor al negocio.

5.4.1.1 Modelo de suscripción

Como parte del modelo de negocio de *Software as a Service*, existen diversas variantes que le agregan más valor al cliente dependiendo del uso que se le dé a la solución. Algunas de estas variantes son discutidas en capítulos anteriores para mayor referencia.

En el presente estudio, se propone un modelo de suscripción que se ajusta más a las necesidades del mercado, considerando que la solución será utilizada por todo tipo de *freelancers*, desde médicos, abogados hasta diseñadores, quienes tienen diferentes capacidades de paga.

Por tal razón, el modelo de suscripción propuesto se enfoca en la cantidad de casos/proyectos que se puedan atender al mes, a través de la solución, de tal manera que si el *freelancer* está iniciando su negocio, muy probablemente la cantidad de casos/proyectos atendidos será pequeña, mientras que un *freelancer*, por ejemplo, un doctor con largo recorrido, la cantidad de casos/proyectos atendidos será alta. Esto le brindaría a cualquier *freelancer* la capacidad de utilizar la solución sin realizar gastos fuera de su alcance.

Por lo tanto, la variable principal de la cual dependerá el modelo de suscripción será: la cantidad de casos por mes. En la siguiente sección se definirá la estrategia de precios tomando como base la variable mencionada.

5.4.1.2. Planes de suscripción

La variable principal sobre la cual se basará el modelo de negocios, es la cantidad de casos atendidos mensualmente, debido a esto se propone un modelo de precios que cubra cuatro posibles escenarios para los clientes, en donde cada uno de ellos posea diferente capacidad de paga.

El objetivo de realizar esta estrategia de precios es captar la mayor cantidad de clientes del mercado, permitiendo a aquellos que se encuentren en sus primeros pasos de *freelancing*, hacer uso de la solución y captarlos como clientes.

Como se ha mencionado en secciones previas, el mercado principal para la solución propuesta en el presente documento, corresponde a aquellos *freelancers* que laboran en el área de medicina y abogacía. Por lo tanto, el almacenamiento de documentos como: recetas, exámenes, ultrasonidos, escrituras, expedientes, entre otros, es de suma importancia para este mercado, ya que le permitiría al profesional tener desde una misma aplicación de *software* todo lo relacionado al historial de sus clientes y cada uno de sus casos/proyectos. Esta sería una segunda variable a considerar dentro de la estrategia de precios.

Freelance

Corresponde a aquel profesional que recientemente inició a trabajar de forma independiente y posee una pequeña cartera de clientes. En esta situación, la cantidad de casos/proyectos que pueda atender el *freelance* a nivel mensual será baja, y por lo tanto, su capacidad de paga probablemente no sea alta.

El plan *Freelance* incluiría desde diez hasta veinte casos/proyectos creados durante un mes, de tal manera que si alcanza ese límite, deberá adquirir el siguiente nivel de los planes ofrecidos.

Adicionalmente, se proporcionaría una capacidad de almacenamiento de documentos de hasta 10GB.

Workteam

Este plan está orientado para aquellos *freelancers* que trabajan en equipo, de tal manera que, puedan distribuir los gastos incurridos en la solución para hacer uso de la misma. La solución brinda la capacidad de limitar la información a la que se tiene acceso, a partir de los casos/proyectos en los que el usuario esté asociado. Siendo así, se podrían crear casos/proyectos para cada uno de los integrantes del equipo de trabajo, y posteriormente asociarlos para su respectiva visualización. Con esta funcionalidad, todos los integrantes del equipo de trabajo podrían hacer uso de la misma solución incurriendo en un único gasto.

Este plan incluiría la creación de veinticinco a cincuenta casos/proyectos mensuales, y de igual manera, si se llega al máximo, se debería de optar por el siguiente nivel de plan. También se incluye una capacidad de almacenamiento de documentos de hasta 25 GB.

Office

El tercer nivel dentro de los planes estaría enfocado para aquellos negocios que ya tienen un recorrido más largo, y cuentan con una base de clientes y casos/proyectos mensuales más grande. De igual manera, haciendo uso de las

funcionalidades de la solución, se podrían asignar roles y permisos que posibiliten una gestión más granular de cada uno de los casos/proyectos creados y de las actividades relacionadas.

El plan *Office* brinda la capacidad de crear desde cincuenta a cien casos/proyectos mensuales, siendo una cantidad ideal para aquellas oficinas o negocios que requieran de una solución que les permita dar un mejor seguimiento a sus diferentes clientes.

Por otro lado, incluye almacenamiento de documentos de hasta 50 GB, puede cargar a la solución cualquier tipo de documento que sea necesario para proporcionar un mejor servicio a sus clientes, y así respaldar el trabajo realizado.

Enterprise

El último nivel propuesto, es el plan *Enterprise* que permite la creación desde cien casos/proyectos en adelante de forma mensual, permitiendo dar seguimiento a suficientes casos como sean necesarios por una empresa.

Este tipo de plan está enfocado para aquellos bufetes de abogados que ya cuentan con una fuerte base de clientes y atienden más de casos mensuales, o bien, para aquellos médicos que llegan a atender hasta ocho citas diarias.

El plan *Enterprise* brinda una capacidad de hasta 100 GB de almacenamiento de documentos.

Cómo se ha mencionado anteriormente, el objetivo de proporcionar estos planes de suscripción, es brindar al cliente las facilidades de adquirir una solución que se adapte a sus necesidades de negocio y a su capacidad de paga conforme su negocio crece.

De esta manera, el cliente puede iniciar con el plan de suscripción *Freelance*, y conforme su negocio se desarrolla puede subir de plan hasta llegar

al plan *Enterprise.* En la Tabla XXVI, se pueden observar los diferentes planes de suscripción propuestos para la solución.

Tabla XXVI. **Planes de suscripción**

	Freelance	Workteam	Office	Enterprise
Precio	USD$ 10	USD$ 25	USD$ 50	USD$ 100
Casos	10 – 25	25 – 50	50 – 100	Más 100
Almacenamiento de documentos	Hasta 10 GB	Hasta 25 GB	Hasta 50 GB	Hasta 100 GB

Fuente: elaboración propia.

5.4.1.3. Servicio de almacenamiento de documentos

Debido al máximo de almacenamiento de documentos que se presenta en los planes de suscripción, se puede crear un servicio adicional orientado a la ampliación de la capacidad de almacenamiento, posibilitando a aquellos clientes que han llegado al límite, adquirir dicho servicio. Este servicio tendría el siguiente plan de suscripción:

Tabla XXVII. **Plan de almacenamiento de documentos**

	Plan de almacenamiento de documentos
Precio	USD$ 5
Capacidad	Hasta 10 GB

Fuente: elaboración propia.

Con este modelo de suscripción, se provee la flexibilidad suficiente para que el *freelancer* sea capaz de optar por el mejor plan que se ajuste a sus necesidades, y adicionar almacenamiento de documentos si fuese necesario. De esta manera, el *freelancer* podría combinar las opciones y diseñar un plan óptimo para su negocio.

6. DISCUSIÓN DE RESULTADOS

En el capítulo anterior, se presentó el diseño de la solución en la nube, haciendo uso de los servicios de *Amazon Web Services,* con el propósito de plasmar una solución robusta que sea capaz de brindar alta disponibilidad a los usuarios. Además de cumplir con diversos estándares de seguridad para certificar que la información de los usuarios se encuentra protegida ante cualquier eventualidad que pueda surgir.

Con base a los lineamientos descritos en la sección anterior, se puede dar respuesta a los objetivos específicos del estudio:

- Diseñar los procedimientos de aprovisionamiento de computación, memoria, *networking* y almacenamiento en disco que permita la implementación inmediata y 100 % funcional de un *software* como servicio.
- Identificar los componentes necesarios para la creación de una infraestructura en la nube que cumpla con estándares de seguridad.
- Definir el modelo de negocio a utilizar que capte la mayor cantidad de usuarios para crear una solución rentable.

6.1. Procedimientos de aprovisionamiento

La solución *Software as a Service* propuesta en el presente documento, debe considerar diversos procedimientos para certificar la disponibilidad de la solución a los usuarios de la misma. Para esto, se describieron en el capítulo anterior, los procedimientos de aprovisionamiento más importantes, para cumplir con dicho objetivo. Los procedimientos de aprovisionamiento cubiertos en el presente estudio son:

- Arquitectura multi-tenant con uso de subdominio

- Evaluación de disponibilidad de subdominio (cliente)
- Asignación de subdominio (cliente)
- Evaluación de recursos computacionales
- Definición de proceso de autoescalamiento

Una de las ventajas de utilizar un proveedor de servicios en la nube, es que se tiene la flexibilidad de incrementar fácilmente la capacidad de los servidores, en cuanto a memoria, espacio en disco, procesador, red y otros servicios, todo esto puede ser en función de la demanda que se tenga en el momento.

Por tal razón, los procedimientos de aprovisionamiento están muy orientados hacia la escalabilidad horizontal de la infraestructura utilizada para la solución, más allá de la asignación de recursos específicos hacia cada uno de los clientes. El propósito de esto, como se presenta en el capítulo anterior, es permitir que la solución *Software as a Service* crezca en su capacidad de forma automatizada, sin importar la cantidad de clientes que se tengan activos o suscritos, sino importando la cantidad de recursos necesarios para suplir la demanda de los clientes que están utilizando la solución.

En la Tabla XXVIII, se presentan los recursos que cubre cada uno de los procedimientos de aprovisionamiento.

Tabla XXVIII. **Recursos evaluados por procedimientos de aprovisionamiento**

Procedimiento	Almacenamiento en disco	Memoria	Computación	*Networking*
Arquitectura multi-tenant con uso de subdominio	✓			

Evaluación de disponibilidad de subdominio (cliente)	✓			
Asignación de subdominio (cliente)	✓			
Evaluación de recursos computacionales		✓	✓	✓
Definición de proceso de autoescalamiento		✓	✓	✓

Fuente: elaboración propia.

Como se puede observar en la Tabla XXVIII, los procedimientos de aprovisionamiento definidos, cubren todas las áreas de recursos evaluados para proveer de una solución óptima en cuanto a su rendimiento.

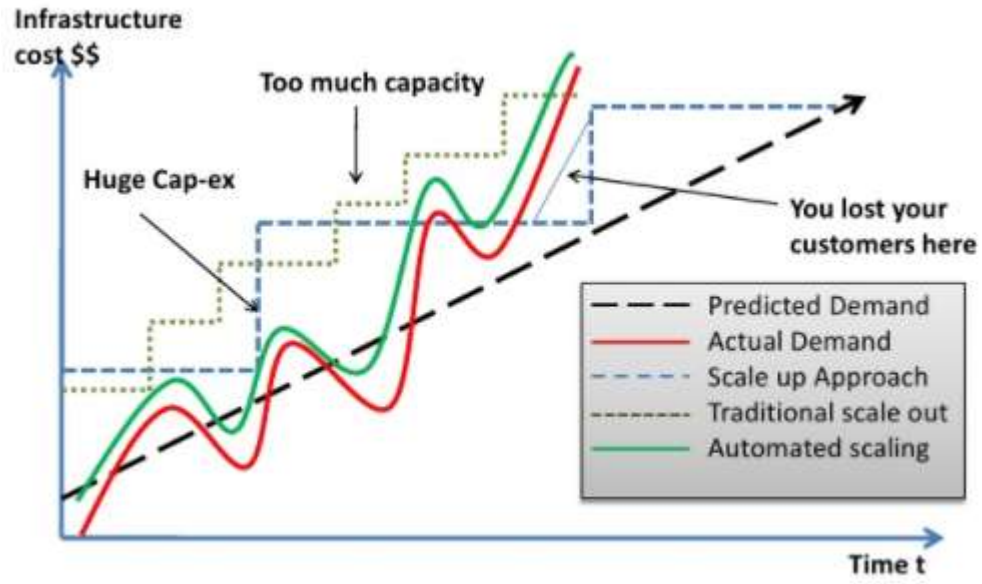
Adicionalmente, el procedimiento definición de proceso de autoescalamiento, se encarga de trabajar de forma proactiva para asegurar alta disponibilidad en el servicio, lo cual cubre parte del objetivo que busca la implementación inmediata y 100% funcional de la solución para un cliente nuevo.

De tal manera que, los servicios configurados de *Amazon Web Services*, estarán constantemente evaluando el uso de los recursos de la solución, para posteriormente proceder con el autoescalamiento si fuese necesario. Siendo así, en todo momento se contaría con un conjunto de servidores que estarían disponibles para atender la demanda del servicio. Y bien, si la evaluación de los

recursos diera como resultado una alerta en cualquiera de los recursos computacionales, el autoescalamiento procedería a incluir servidores adicionales a la infraestructura, lo cual involucraría más memoria, más procesamiento computacional y *networking*, mejorando el rendimiento de la solución.

En la Figura 29, se puede observar el comportamiento del autoescalamiento respecto a la demanda.

Figura 29. **Autoescalamiento Amazon AWS**



Fuente: *Amazon AWS.*

Como se observa en la Figura 29, el autoescalamiento en la nube permite reducir los costos de infraestructura en comparación de los modelos tradicionales de escalamiento. En donde dichos modelos, podían ser: escalamiento horizontal o escalamiento vertical, y en cualquiera de estos casos, el costo de inversión era mayor en algún punto de la prestación del servicio, o bien, no se cubría completamente la demanda, entregando un servicio degradado en su rendimiento.

Por otro lado, el autoescalamiento responde de forma óptima a ambas variables: costos y capacidad, sin afectar la demanda actual, por el contrario, ajustándose automáticamente a ella.

6. 2. Controles de seguridad

De acuerdo a lo revisado en el capítulo anterior, la seguridad es un pilar de suma importancia en cuanto al uso de una solución *Software as a Service*. Por tal razón, se han detallado en el presente documento los diferentes controles de seguridad que, como mínimo, deben ser considerados durante la implementación de una solución de este tipo. Estos controles de seguridad cubren diversos aspectos:

- Confidencialidad: es la propiedad de la información que garantiza que la información es únicamente accesible por aquellas personas que han sido correctamente autorizadas.
- Integridad: es la propiedad de la información que garantiza que la misma no ha sufrido de alteraciones o modificaciones sin la autorización correspondiente. Esto permite asegurar que la información a que se tiene acceso, es válida y legítima.
- Disponibilidad: es la propiedad de la información que asegura su accesibilidad en todo momento a los usuarios de la misma.

En la Tabla XXIX, se pueden observar los controles de seguridad discutidos en el capítulo anterior, y los aspectos que se cubren.

Tabla XXIX. **Aspectos cubiertos por los controles de seguridad**

Control de seguridad	Confidencialidad	Integridad	Disponibilidad
Comunicación HTTPS	✓	✓	
Autenticación	✓		

Control de acceso basado en roles	✓		
Variables de sesión	✓		
Web application firewall		✓	✓
Subredes		✓	
Firewall		✓	✓
Tabla de ruteo		✓	✓
Encriptación de base de datos	✓	✓	
Replicación de base de datos		✓	✓
Respaldos de base de datos		✓	✓
Política de privacidad	✓		

Fuente: elaboración propia.

Con base al anterior resumen, se puede concluir que los controles propuestos como parte del diseño de la solución, cubren las principales propiedades de la información.

6.3. Rentabilidad

En el capítulo anterior, se define el modelo negocio y los planes de suscripción de la solución *Software as a Service*. Adicional a dicha definición, en esta sección se realizan las estimaciones correspondientes para confirmar la rentabilidad del negocio, ya que este puede asegurar montos altos de ventas,

pero a su vez, los costos de ventas en los que se incurre pueden ser también altos, generando un pequeño margen de ganancia, y por lo tanto, baja rentabilidad.

En este estudio, se presentan diferentes escenarios para confirmar que el modelo de negocio es rentable, lo cual se hace, a partir de dos análisis: margen bruto y tasa interna de retorno.

6.3.1. Margen neto

En este primer análisis, se identifica el margen bruto del modelo de negocios, que es el resultado de:

Margen bruto = Ventas - Costos

El objetivo de este análisis corresponde a la necesidad de determinar si la solución cumple con el modelo de economías de escala respecto a las ventas. Dicho modelo presenta la ventaja de incrementar la producción mientras los costos y gastos de dicha producción se mantienen iguales o se reducen, generando un alto margen de ganancia, que es lo que se busca en una solución *Software as a Service.*

Este análisis ayuda a dilucidar la ventaja que representa la creación o implementación de una solución *Software as a Service,* enfocándose únicamente en las ventas y costos de ventas, ya que son indicadores que están directamente relacionados y normalmente son proporcionales.

En la Tabla XXX, se puede observar el escenario en el cual se consideran todos los aspectos técnicos necesarios para proveer la solución, una vez ya se tienen ingresos como parte de las suscripciones. Los costos de ventas en los que se incurre para la entrega de la solución serían:

- Base de datos
- Servidores

- Ruteo
- Almacenamiento
- Dominio
- Certificado de seguridad
- Transferencia de datos
- Alarmas
- Balanceador de carga
- Zendesk

Tabla XXX. **Margen bruto, año 1 (USD $)**

Año 1	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7	Mes 8	Mes 9	Mes 10	Mes 11	Mes 12
Ingresos totales	**100**	**200**	**300**	**400**	**600**	**800**	**1000**	**1250**	**1500**	**1800**	**2250**	**2750**
Office	**100**	**200**	**300**	**400**	**600**	**800**	**1000**	**1250**	**1500**	**1800**	**2250**	**2750**
Suscriptores activos	2	4	6	8	12	16	20	25	30	36	45	55
Costos totales	**962**	**35**	**38**	**41**	**50**	**56**	**208**	**81**	**229**	**104**	**261**	**283**
Base de datos	481											
Servidores	137	-	-	-	-	-	137.00	-	137.00	-	137.00	137.00
Ruteo	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Almacenamiento	2.5	5.0	7.5	10.0	15.0	20.0	25.0	31.3	37.5	45.0	56.3	68.8
Dominio	12											
Certificado de seguridad	300											
Transferencia de datos	0.6	1.2	1.8	2.4	3.6	4.8	6.0	7.5	9.0	10.8	13.5	16.5
Alarmas	3	3	3	3	6	6	9	12	15	18	24	30
Balanceador de carga	20	20	20	20	20	20	20	20	20	20	20	20
Zendesk	5	5	5	5	5	5	10	10	10	10	10	10
Margen bruto	(862)	165	262	359	550	744	793	1,169	1,271	1,696	1,989	2,467
Margen bruto acumulado	**(862)**	**(696)**	**(434)**	**(75)**	**475**	**1,219**	**2,011**	**3,180**	**4,451**	**6,147**	**8,135**	**10,603**
% Margen bruto	**(861.60)**	**82.65**	**87.40**	**89.78**	**91.65**	**92.96**	**79.25**	**93.50**	**84.73**	**94.21**	**88.39**	**89.72**

Fuente: elaboración propia.

La Tabla XXX como base para el análisis correspondiente del margen bruto, despliega que el porcentaje del mismo se mantiene entre 85 y 90 %, lo que permite confirmar que el modelo de negocio si presenta las ventajas de las economías de escala, a mayores ventas se mantienen los costos de ventas.

6.3.2. Tasa interna de retorno

Al iniciar un negocio, es idóneo contar con el suficiente apoyo económico que faculte a los dirigentes para desarrollarlo de la mejor manera, considerando diferentes factores que puedan influir en el éxito del mismo. Como parte de esos factores, se deben estimar:

- Aspectos técnicos (costos):
    - Los cuales se detallan en la sección previa
- Capital humano para (gastos):
    - Desarrollo, entrega y soporte técnico de la solución
    - Comercialización de la solución
    - Promoción y mercadeo la solución

El capital humano representa un alto porcentaje de los gastos en los que pueda incurrir una empresa, debido a que este representa el motor de todo negocio y se encarga de las operaciones de la misma. Sin capital humano, la empresa no puede operar, por ende, no se lograrían los objetivos planteados.

Por esta razón, es importante considerar un escenario en el cual se tenga el apoyo económico de un inversionista que realice un aporte inicial para dar comienzo con las operaciones del negocio.

Suponiendo que se cuenta con una inversión inicial de USD 50,000, se pueden realizar las siguientes estimaciones.

Tabla XXXI. **Estado de resultados proyectado, año 1 (USD $)**

Año 1												
	Mes 1	**Mes 2**	**Mes 3**	**Mes 4**	**Mes 5**	**Mes 6**	**Mes 7**	**Mes 8**	**Mes 9**	**Mes 10**	**Mes 11**	**Mes 12**
Ingresos	-	-	-	100	200	300	400	600	800	1,000	1,250	1,500
Costos	8	8	8	962	35	38	41	50	56	208	81	229
Margen bruto	(8)	(8)	(8)	(862)	165	262	359	550	744	793	1,169	1,271
Gastos	1,300	1,300	1,300	1,300	1,355	1,340	2,605	1,940	1,955	2,590	1,955	1,940
Utilidad neta	(1,308)	(1,308)	(1,308)	(2,162)	(1,190)	(1,078)	(2,246)	(1,390)	(1,211)	(1,798)	(786)	(669)
Utilidad neta acumulada año	(1,308)	(2,616)	(3,924)	(6,086)	(7,276)	(8,354)	(10,599)	(11,990)	(13,201)	(14,998)	(15,785)	(16,454)
Utilidad neta acumulada	**(1,308)**	**(2,616)**	**(3,924)**	**(6,086)**	**(7,276)**	**(8,354)**	**(10,599)**	**(11,990)**	**(13,201)**	**(14,998)**	**(15,785)**	**(16,454)**

Fuente: elaboración propia.

Tabla XXXII. **Estado de resultado proyectado, año 2 (USD $)**

Año 2												
	Mes 1	**Mes 2**	**Mes 3**	**Mes 4**	**Mes 5**	**Mes 6**	**Mes 7**	**Mes 8**	**Mes 9**	**Mes 10**	**Mes 11**	**Mes 12**
Ingresos	1,800	2,250	2,750	3,500	4,250	5,000	5,750	6,500	7,250	8,000	8,750	9,500
Costos	104	261	283	2,010	476	366	395	561	453	625	517	546
Margen bruto	1,696	1,989	2,467	1,490	3,774	4,635	5,355	5,939	6,797	7,376	8,233	8,954
Gastos	3,205	2,540	2,555	3240	4255	4240	4255	4240	4255	4240	5940	5940
Utilidad neta	(1,509)	(551)	(88)	(1,750)	(481)	395	1,100	1,699	2,542	3,136	2,293	3,014
Utilidad neta acumulada año	(1,509)	(2,061)	(2,148)	(3,898)	(4,380)	(3,985)	(2,885)	(1,186)	1,356	4,491	6,785	9,799
Utilidad neta acumulada	**(17,963)**	**(18,514)**	**(18,602)**	**(20,352)**	**(20,833)**	**(20,439)**	**(19,338)**	**(17,639)**	**(15,098)**	**(11,962)**	**(9,669)**	**(6,655)**

Fuente: elaboración propia.

Tabla XXXIII. **Estado de resultados proyectado, año 3 (USD $)**

Año 3												
	Mes 1	**Mes 2**	**Mes 3**	**Mes 4**	**Mes 5**	**Mes 6**	**Mes 7**	**Mes 8**	**Mes 9**	**Mes 10**	**Mes 11**	**Mes 12**
Ingresos	11,500	12,000	13,500	14,000	15,000	16,750	17,500	18,750	21,000	21,250	22,500	24,750
Costos	876	781	1,061	5,229	939	1,123	1,034	1,082	1,130	1,324	1,235	1,283
Margen bruto	10,625	11,220	12,440	8,772	14,062	15,627	16,466	17,668	19,871	19,926	21,265	23,467
Gastos	5955	5940	5955	5940	8555	8540	8555	8540	8555	8540	11540	11540
Utilidad neta	4,670	5,280	6,485	2,832	5,507	7,087	7,911	9,128	11,316	11,386	9,725	11,927
Utilidad neta acumulada año	4,670	9,949	16,434	19,265	24,772	31,858	39,769	48,898	60,213	71,599	81,324	93,251
Utilidad neta acumulada	**(1,985)**	**3,294**	**9,779**	**12,610**	**18,117**	**25,203**	**33,114**	**42,243**	**53,558**	**64,944**	**74,669**	**86,596**

Fuente: elaboración propia.

Es importante considerar que la creación de un negocio requiere de cierto esfuerzo adicional, con el objetivo de reducir gastos innecesarios en una etapa inicial, lo cual permite al negocio dar sus primeros pasos sin afectar su solvencia económica o los intereses de un inversionista.

Los gastos incurridos para el primer año, se detallan en la Tabla XXXIV que se puede observar a continuación.

Tabla XXXIV. **Gastos proyectados, año 1**

Tipo de Gasto	Gastos	Cantidad	Meses
Capital humano	Vendedores	1	6
	Desarrolladores de sistemas	2	12
	Diseñador/Ejecutivo de mercadeo	1	2
Publicidad	Publicidad en redes sociales		8

Fuente: elaboración propia.

Para el segundo año de operación, se tienen los gastos de la Tabla XXXV.

Tabla XXXV. **Gastos proyectados, año 2**

Tipo de Gasto	Gastos	Cantidad	Meses
Capital humano	Vendedores	2	12
	Supervisor de ventas	1	2
	Desarrolladores de sistemas	2	12
	Analista/Desarrollador	1	8
	Soporte técnico	1	2
	Diseñador/Ejecutivo de mercadeo	1	12
Publicidad	Publicidad en redes sociales		12

Fuente: elaboración propia.

En el segundo año, ya se consideran recursos adicionales para la operación, tanto tecnológica como del negocio, en donde uno de los objetivos es empezar a expandirse mediante el incremento en las ventas y la inclusión

permanente de un ejecutivo de mercado para la gestión de publicidad en redes sociales. Para el tercer año de operación, se consideran los gastos de la Tabla XXXVI.

Tabla XXXVI. **Gastos proyectados, año 3**

Tipo de Gasto	Gastos	Cantidad	Meses
Capital humano	Vendedores	2	12
	Supervisor de ventas	1	12
	Desarrolladores de sistemas	2	12
	Analista/Desarrollador	2	12
	Jefe de desarrollo	1	2
	Soporte técnico	2	12
	Ejecutivo de mercadeo	1	8
	Diseñador/ Ejecutivo de mercadeo	1	14
Publicidad	Publicidad en redes sociales		12

Fuente: elaboración propia.

El capital humano es clave para el desarrollo exitoso de cualquier negocio, por ende, la inversión en dicho rubro es importante y requiere de constantes mejoras. En este caso, se presenta el escenario en donde cada una de las áreas del negocio como: tecnología, ventas y mercadeo, aumentan la cantidad de involucrados permitiendo entregar una solución más dinámica tecnológicamente, y expandirse a otros mercados.

A pesar de requerir un mayor gasto durante el segundo y tercer año de operación, los resultados son positivos de acuerdo a los ingresos obtenidos, lo cual confirma que el negocio se vuelve rentable después de cierto tiempo de operación. A este punto en donde los ingresos acumulados son igual que los egresos acumulados, se le llama punto de equilibro. El escenario anterior se puede observar en la Tabla XXXIII, que el punto de equilibro se encuentra en el mes 26.

Posterior a dicho mes, los resultados tienden a ser positivos, tanto acumulados como mensuales. Este indicador es clave para determinar cuándo un negocio empieza a ser rentable.

Por otro lado, también se requiere el cálculo de otro importante indicador, llamado: tasa interna de retorno. Este indicador posibilita a todo negocio identificar la tasa de interés con la cual se recuperará el capital invertido, de tal manera que, todo inversionista solicita dicho indicador para determinar si el negocio le es viable o no. El cálculo de la tasa interna de retorno, se puede observar en la Tabla XXXVII.

Tabla XXXVII. **Cálculo de TIR al año 3**

Inversión inicial	**(USD $)**	**(47,000)**		**TIR**	
Ingresos 1 año	(USD $)	(16,454)		TIR (1 año)	-
Ingresos 2 año	(USD $)	9,799		TIR (2 años)	-69%
Ingresos 3 año	(USD $)	93,251		TIR (3 años)	20%

Fuente: elaboración propia.

Como del cálculo de la tasa interna de retorno, se puede observar que al tercer año, se tiene una tasa de 20 %, queriendo decir que al tercer año ya se habrá recuperado en su totalidad el capital invertido inicialmente. Esto permite concluir que el modelo de negocio es rentable.

Por otro lado, se logra determinar que el punto de equilibrio del negocio, se da en las siguientes condiciones:

- 240 clientes con plan de suscripción Enterprise.
- O su equivalente a USD $ 12,000 mensuales de ingresos, en cualquier combinación de planes de suscripción.
- Al llegar al mes 26 de estar en operación.

CONCLUSIONES

Con base al diseño propuesto en el presente estudio, se puede llegar a las siguientes conclusiones:

1. Con el uso de diversas tecnologías en la nube, como *Amazon Web Services*, en combinación con procedimientos de aprovisionamiento, controles de seguridad y un modelo de negocio rentable, se concluye que sí se puede diseñar una solución *Software as a Service,* que permita el despliegue de un sistema de gestión de operaciones para *freelancers* de forma inmediata y confiable.

2. Los procedimientos de aprovisionamiento definidos en el estudio son: evaluación de disponibilidad de subdominio, asignación de subdominio, evaluación de recursos computacionales y proceso de autoescalamiento. Con el uso de estos procedimientos, se logra proveer de los servicios tecnológicos requeridos para entregar de forma inmediata y completamente funcional la solución a los clientes, y adicionalmente se determina que aseguran un servicio ininterrumpido, constante y estable para satisfacer la creciente demanda.

3. Los controles de seguridad que forman parte del diseño propuesto son: uso de protocolo de comunicación seguro (HTTPS), autenticación, control de acceso basado en roles, variables de sesión, *web application firewall,* subredes para cada grupo de servidores, *firewall* a nivel de red, tablas de ruteo, encriptación, replicación y respaldo de base de datos. Con el uso de estos controles de seguridad, aunados a una política de privacidad para asegurar la confidencialidad de la información, se concluye que brindan una

solución que cumpla con diversos estándares de seguridad para ganar la confiabilidad de los clientes.

4. El modelo de negocio incluye puntos clave para la captación de la mayor cantidad de clientes posibles, como: los planes de suscripción y los servicios adicionales que pueden agregar valor a la solución. Teniendo esto como base, en conjunto con los egresos en los que se incurriría para prestar el servicio, se puede determinar que el modelo propuesto es rentable, sin embargo, se requiere de una inversión inicial que sea capaz de poner en operación el negocio.

RECOMENDACIONES

1. La solución descrita en el presente estudio está basada en el uso de tecnologías de *Amazon Web Services* para los recursos de infraestructura en la nube. Considerando que hoy en día existen diferentes proveedores de infraestructura en la nube como Microsoft, Google, IBM, entre otros, se recomienda realizar un análisis detallado sobre las ventajas de alojar la solución en uno u otro proveedor, desde el punto de vista tecnológico y financiero.

2. La seguridad es un pilar fundamental en toda solución, principalmente cuando se trata de un *Software as a Service*. En el presente estudio se analizaron diferentes niveles de seguridad, desde los servidores hasta el acuerdo la política de privacidad. Sin embargo, para que una solución *Software as a Service* transmita mayor confianza a sus clientes, sería ideal contar con diferentes programas de cumplimiento, tales como: CSA (Cloud Security Alliance Controls), ISO 9001 (Global Quality Standard), PCI (Payment Card Standards), entre otros. Por lo tanto, se recomienda evaluar el costo de dichos programas de cumplimientos y su impacto en la rentabilidad de la solución.

3. El modelo de negocio propuesto en el presente estudio, se enfoca fundamentalmente en un plan de suscripción acorde a la cantidad de casos (proyectos) del *freelancer*; sin embargo, se recomienda la evaluación de otros planes que pudiesen llegar a generar mayor captación de clientes, y que a su vez generen más ingresos. En este sentido, se recomienda que se evalúe al menos la inclusión de planes semestrales y/o

anuales. Adicionalmente, se recomienda evaluar la oferta de otros servicios tales como: firma electrónica, facturación electrónica, entre otros.

REFERENCIAS BIBLIOGRÁFICAS

Anthony, R., Blau, J., & Princeton Review (Firm). (2002). *Job Surfing: Freelancing : Using the Internet to Find a Job and Get Hired.* Massachusetts, Estados Unidos: The Princeton Review.

Bloomberg, J. (2013). *The Agile Architecture Revolution: How Cloud Computing, REST-Based SOA, and Mobile Computing Are Changing Enterprise IT.* Nueva Jersey, Estados Unidos: Wiley.

Bouzid, A., & Rennyson, D. (2015). *The Art of SaaS.* Indiana, Estados Unidos: Xlibris Corporation.

Doriwala, J. (2014). *Freelancing-TheFutureOfWork: Start Working from Any Part of The World with a Computer and Internet.* Surat, India: TenaciousTechies.

Greer, M. B. (2009). *Software as a Service Inflection Point: Using Cloud Computing to Achieve Business Agility.* Indiana, Estados Unidos: iUniverse.

Hossain, S. (2012). Cloud Computing Terms, Definitions, and Taxonomy. En A. M. Bento, & A. K. Aggarwal, *Cloud Computing Service and Deployment Models: Layers and Management: Layers and Management* (págs. 1-25). Pensilvania, Estados Unidos: IGI Global.

Kizza, J. M. (2015). *Guide to Computer Network Security.* Berlín, Alemania: Springer.

Lepofsky, R. (2014). *The Manager's Guide to Web Application Security: A Concise Guide to the Weaker Side of the Web.* Nueva York, Estados Unidos: Apress.

Mather, T., Kumaraswamy, S., & Latif, S. (2009). *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance.* Massachusetts, Estados Unidos: O'Reilly Media, Inc.

Mujawar, T. N., Sutagundar, A. V., & Ragha, L. L. (2017). Security Aspects in Cloud Computing. En N. K. Kamila, *Advancing Cloud Database Systems and Capacity Planning With Dynamic Applications* (págs. 320-342). Pensilvania, Estados Unidos: IGI Global.

Prabowo, A. H., Janssen, M., & Barjis, J. (2012). A Conceptual Model for Assessing the Benefits of *Software* as a Service from Different Perspectives. En W. Abramowicz, D. Kriksciuniene, & V. Sakalauskas, *Business Information Systems* (págs. 108-119). Berlín, Alemania: Springer.

Rao, M. N. (2015). *Cloud Computing.* Delhi, India: PHI Learning Pvt. Ltd.

Shema, M. (2012). *Hacking Web Apps: Detecting and Preventing Web Application Security Problems.* Oxford, Newnes: Reino Unido.

Strader, T. J. (2010). *Digital Product Management, Technology and Practice: Interdisciplinary Perspectives.* Pensilvania, Estados Unidos: IGI Global.

Tipton, H. F., & Krause, M. (2007). *Information Security Management Handbook.* Florida, Estados Unidos: CRC Press.

Vasudevan, V. (2008). *Application Security in the ISO27001 Environment.* Cambridge, Reino Unido: IT Governance Ltd.

Zhu, W.-D., Benoit, B., Jackson, B., Liu, J., Marin, M., Meena, S., . . . IBM Redbooks. (2015). *Advanced Case Management with IBM Case Manager.* Nueva York, Estados Unidos: IBM Redbooks.

Zhu, Y., Zhang, J., & Li, J. (2012). Solution for Transforming Web Application to Multi-tenant SaaS Application. En Y. Zhang, *Future Communication, Computing, Control and Management, Volume 1* (págs. 181-188). Berlín, Alemania: Springer Science & Business Media.

ANEXOS

Política de privacidad

Política de privacidad, efectiva a partir de <día> de <mes>, <año>

<Empresa> (“Empresa”, “nosotros”, o la “Compañía”) respeta la necesidad de privacidad en línea y protege de manera apropiada la información de aquellos que visitan (“Visitantes”) y que se registran (“Clientes”) en sus sitios.

La práctica utilizada por empresa respecto al uso de la información personal de sus clientes es como se expone a continuación en esta política de privacidad. Como condición para utilizar los servicios proporcionados por empresa, el cliente acepta los términos de esta política de privacidad y está consciente que la misma puede ser actualizada periódicamente.

Alcance

Esta política de privacidad se aplica exclusivamente a www.<empresa>.com, sitio en el cual un visitante puede suscribirse como cliente para hacer uso del servicio que se presta por empresa. El servicio (“Servicio”) está enfocado específicamente en la gestión de operaciones que el cliente puede realizar para satisfacer la demanda de sus clientes.

Adicionalmente, en esta política se incluye la forma en que empresa usa y asegura la información proveída por sus clientes a través del servicio, ya sea mediante el ingreso de información como datos, documentos o mensajes digitales compartidos (“Información”) entre los usuarios.

Información de nuestros clientes

Durante el proceso de suscripción se solicita a los clientes su nombre, dirección de correo electrónico, contraseña y nombre de cuenta (“Cuenta”) sobre la cual se utilizará el servicio, información que se utilizará para comunicarle sobre nuevas

funcionalidades, lanzamientos, disponibilidad del servicio y cambios en esta política de privacidad.

En la Empresa almacenaremos la Información proporcionada por nuestros clientes como parte de la utilización de nuestro servicio, tales como nombres de terceros, direcciones de correo electrónico, números de teléfono, direcciones físicas o fiscales, actividades, gastos incurridos, documentos confidenciales, saldos de clientes, cobros, entre otros. La información la almacenaremos como parte del uso exclusivo del servicio adquirido por nuestros clientes. Sin embargo, en empresa no revisamos, compartimos, distribuimos o referenciamos la información proporcionada por nuestro cliente con ningún tercero para ningún fin que no haya sido consentido y que no forme parte del servicio prestado, a excepción de lo requerido por la ley.

La información almacenada por los clientes respecto a sus contactos involucrados en sus operaciones, no es accesible por nuestros empleados en Empresa, ya que se cuentan con políticas de seguridad que lo impiden. Por lo tanto, si existe información personal perteneciente a un tercero que haya sido ingresada por uno de nuestros clientes, y el tercero desea que se modifique o elimine de nuestro almacenamiento, la solicitud deberá realizarse directamente a nuestro cliente, quien es el único capaz de acceder a la misma para cualquier actualización, debido a que en empresa tenemos acceso limitado y restringido a toda la información de nuestros clientes.

Información recolectada

Para recolectar información correspondiente al uso de la solución por parte de los clientes, empresa utiliza diversas técnicas acorde al tipo de información:

Archivos de registro

Los detalles de uso de nuestro servicio por parte de nuestros clientes, como el tiempo, la frecuencia, la duración, el patrón de uso, las funciones utilizadas y la

cantidad de almacenamiento utilizado, serán recolectados por empresa con el fin de ofrecer una constante mejora en la experiencia de nuestro servicio, y a la vez, para entender con mayor profundidad aquellas funcionalidades que son más significativas para nuestros clientes, y así, ofrecer mejoras en las mismas o funciones complementarias que permitan satisfacer otras necesidades.

Otros detalles guardados en nuestros archivos de registro son:

- Direcciones IP
- URL's de las páginas visitadas
- Navegador utilizado
- Dispositivo desde el cual es accedida la solución
- Ubicación: país desde el cual se accede
- Proveedor de servicio de internet
- Búsquedas

Cookies

Empresa también utiliza cookies para almacenar las preferencias del usuario permitiendo brindar una solución más apegada a su realidad optimizando la navegación dentro de la aplicación. Algunas de las cookies almacenadas son aquellas relacionadas a las preferencias del diseño de la solución: color, texto, fuentes, notificaciones, entre otras.

Adicionalmente, se almacenan ciertas características de los permisos del usuario con el propósito de brindar acceso a aquellas áreas seguras dentro de la solución, y así, certificar que dichas áreas únicamente son accedidas por los usuarios a quienes se les han concedido los permisos correspondientes. Esto permite disminuir el tiempo de respuesta de la aplicación, incrementando el rendimiento de la misma y mejorando la experiencia del usuario.

Copias de seguridad

Con el fin de evitar la pérdida de datos, debido a errores o fallas en el sistema, Empresa realiza copias de seguridad de la información que nuestros clientes almacenan en nuestro servicio. Aseguramos que el contenido de su cuenta de usuario no será revelado a nadie y no será accesible incluso a los empleados de empresa excepto en circunstancias específicamente mencionadas en la presente política de privacidad.

Seguridad de la información

Para empresa la seguridad de la información de nuestros clientes, es muy importante. Durante el uso de nuestro servicios, empresa encripta la transmisión de la información, a través de internet mediante una capa de tecnología de seguridad (SSL).

Empresa utiliza diversos estándares de seguridad para cada una de las tecnologías de almacenamiento y procesamiento de datos utilizados como parte del servicio. De igual manera, empresa posee de técnicas de control de acceso para proteger la información de aquellos no autorizados, impidiendo la alteración, divulgación o destrucción de la información.

El acceso a la información de nuestros clientes, está restringido para aquellos de nuestros empleados que necesitan conocerla como parte de la provisión de nuestro servicio, al mismo tiempo que ellos están obligados a cumplir con nuestro acuerdo de confidencialidad y política de privacidad. Estas medidas de seguridad ayudan a prevenir el acceso no autorizado, a mantener la integridad de la información y a asegurar el correcto uso de la misma.

Como parte del servicio, la solución provee a los clientes la capacidad de creación de usuarios, quienes serán capaces de ingresar a la cuenta del cliente y realizar las operaciones correspondientes a los permisos concedidos por el administrador de la cuenta. Por lo tanto, el cliente es el responsable de mantener una correcta

configuración de seguridad de sus usuarios, para certificar que se mantiene la confidencialidad e integridad de su información.

Proveedores

Empresa utiliza terceras partes (“Proveedores”), para proporcionar nuestro servicio. Dichos proveedores brindan el hardware, infraestructura de red, almacenamiento y *software* relacionado para ejecutar nuestro servicio adecuadamente. Los fines para los cuales compartamos los datos de nuestros clientes a nuestros proveedores de servicios, están limitados a almacenamiento de datos, gestión de base de datos y análisis *web*.

Estos proveedores están autorizados a utilizar su información sólo para proporcionar estos servicios a nosotros. Empresa asegura que dichos proveedores de servicios y socios de negocios cumplen con esta política de privacidad y adoptan medidas de confidencialidad y de seguridad apropiadas.

Empresa ratifica que no vende la información de sus clientes a terceros, incluyendo en ellos a sus proveedores.

Investigación de actividad ilegal

Empresa podría divulgar la información de alguno de los clientes a las autoridades de ley correspondientes, con el propósito de:

- Investigar cualquier presunta actividad ilegal o potencial violación de los términos y condiciones de uso de nuestro servicio,
- Proteger la seguridad de nuestros clientes y empleados.

Actualización y eliminación de información

Empresa almacena la información de nuestros clientes durante el tiempo en que se presta el servicio, y un tiempo razonable posterior a la finalización de la suscripción de la cuenta, con el único propósito de responder a intereses de negocio internos como auditorías u obligaciones legales, o bien, para resolver

disputas o para cumplimiento de nuestros acuerdos. Posterior a esto, empresa será responsable de eliminar toda información relacionada de nuestro cliente, siendo esta cualquiera de la detallada en la presente política de privacidad.

Si alguno de nuestros clientes llegase a requerir de alguna modificación a la información que haya sido previamente proporcionada, a través de nuestra solución, puede realizar una solicitud para su correspondiente visualización, actualización y/o eliminación. Dicha solicitud será gestionada en un tiempo considerable de 30 días calendario, durante el cual la información será almacenada en su estado inicial, sin ser alterada por empresa.

Cambios a la política de privacidad

Empresa se reserva el derecho de modificación a esta política de privacidad, sin embargo, se compromete a notificar con al menos 30 días de anticipación mediante nuestro sitio *web* y correo electrónico, sobre cualquier modificación a la misma, permitiendo a nuestros clientes conocer los cambios previo a que se haga efectiva.

La continuidad en el uso de nuestro servicio, establece la aceptación de la política de privacidad, consintiendo los lineamientos constituidos en la misma.

En dado caso, el cliente o visitante no esté de acuerdo con nuestra política de privacidad, la única forma de no aceptarla será mediante la interrupción del uso del servicio por cualquier de los usuarios de la cuenta.

Contacto

Si tiene alguna pregunta o duda sobre esta política de privacidad, por favor contacte con nosotros en legal@empresa.com. Se responderá a todas las consultas dentro de los 30 días siguientes a la recepción de la consulta.